latent: an R library for Latent Variable Modeling

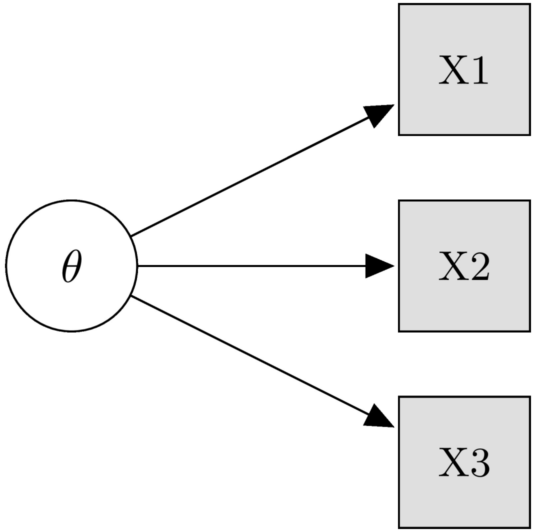
Marcos Jimenez, Mauricio Garnier-Villarreal, & Vithor Rosa Franco

2025-02-05

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

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

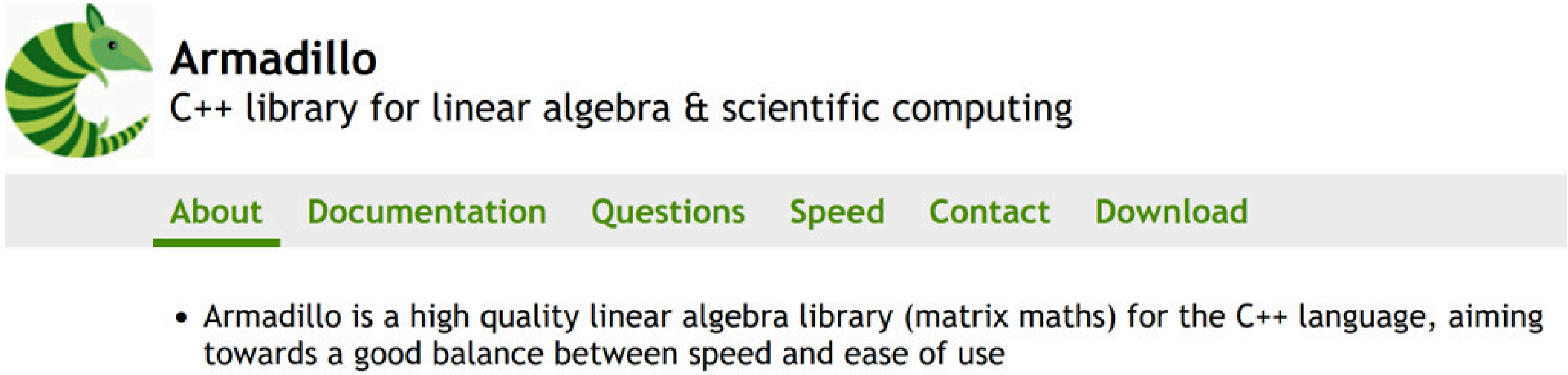
Console | Terminal | Background Jobs

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/
```

High-performance computing

- Core functions written in C++ with the armadillo library



- Parallelization of multiple random starts to address local maxima

Latent Class Analysis

Latent class analysis (LCA) is an umbrella term that refers to a number of techniques for estimating unobserved group membership based on a parametric model of one or more observed indicators of group membership.

People belong to different groups (i.e., classes) that are not directly observable. There are two types of parameter:

- The probability that a person belongs to a particular class k: $P(\theta_k)$.
- The conditional probability (density) of a response to item j if a person belongs to the class k: $f(y_j \mid \theta_k)$.

The likelihood of a response pattern is given by

$$\ell \ = \ \sum_{k=1}^{K} P(\theta_k) \ \prod_{j=1}^{J} f(y_j \mid \theta_k).$$

Categorical indicators example

For this analysis, we will employ the **gss82** example data set included in the `latent` package. Sourced from the `poLCA` package, this data comes from 1,202 respondents to the 1982 General Social Survey.

Model fitting:

▶ Code

```
latent 0.1.0 converged after 66 iterations

  Estimator                                      Penalized-ML
  Optimization method                            lbfgs
  Number of model parameters                     20

  Number of observations                         1202

  Number of response patterns (include NA)       33

  Number of possible patterns                    35

  ------------------------------------------------------

Model Test User Model:
  Test statistic (L2)                            22.088
```

Categorical indicators example

Model fitting:

▶ Code

Model information:

▶ Code

Model Fit indices

The `getfit` function is used for extracting several fit indices for the model.

▶ Code

```
     nclasses              npar              nobs           loglik1 
        3.000            20.000          1202.000         -2754.643 
      loglik2 penalized_loglik                L2               dof 
     2754.643         -2759.507            22.088            15.000 
       pvalue              AIC1              AIC2              BIC1 
        0.106          5549.287         -5469.287          5651.122 
         BIC2             AIC31             AIC32             CAIC1 
    -5367.452          5569.287         -5449.287          5671.122 
        CAIC2              KIC1              KIC2            SABIC1 
    -5347.452          5572.287         -5446.287          5587.594 
       SABIC2              ICL1              ICL2              AICp 
    -5430.980         -6011.741          5006.833          5559.013 
         BICp             AIC3p             CAICp              KICp 
     5660.848          5579.013          5680.848          5582.013 
       SABICp              ICLp        R2_entropy 
     5597.320         -6021.467             0.580 
```

Profile table

The profile table contains the model parameter estimates.

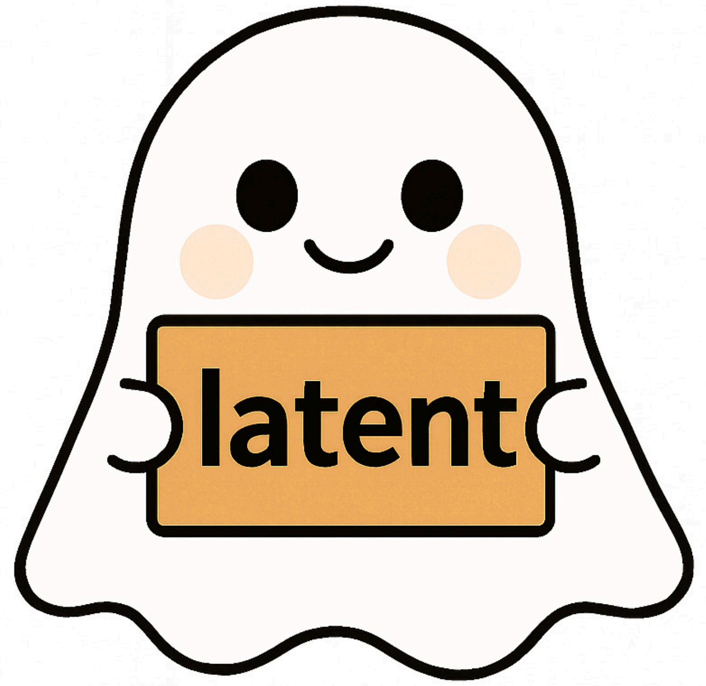
▶ Code

```
$class
Class1 Class2 Class3 
 0.179  0.617  0.204 

$item
$item$PURPOSE
              Class1 Class2 Class3
Good           0.159  0.891  0.916
Depends        0.222  0.052  0.071
Waste of time  0.619  0.057  0.014

$item$ACCURACY
            Class1 Class2 Class3
Mostly true  0.043  0.615  0.653
Not true     0.957  0.385  0.347
```

Continuous indicators example

Model fitting:

▶ Code

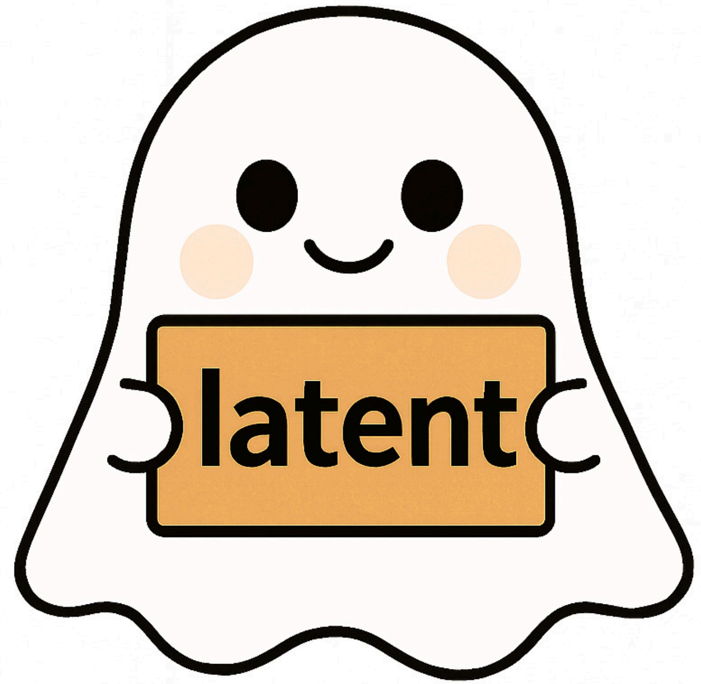


Continuous indicators example

Profile output:

▶ Code

```
$class
Class1 Class2 Class3 Class4
 0.416  0.184  0.153  0.248

$item
$item$ec1
      Class1 Class2 Class3 Class4
Means  2.694  2.249  3.007  2.458
Stds   0.451  0.466  0.532  0.458

$item$ec2
      Class1 Class2 Class3 Class4
Means  3.086  2.173  3.389  2.717
Stds   0.527  0.505  0.541  0.261

$item$ec3
```



Mixed indicators example

Model fitting:

▶ Code

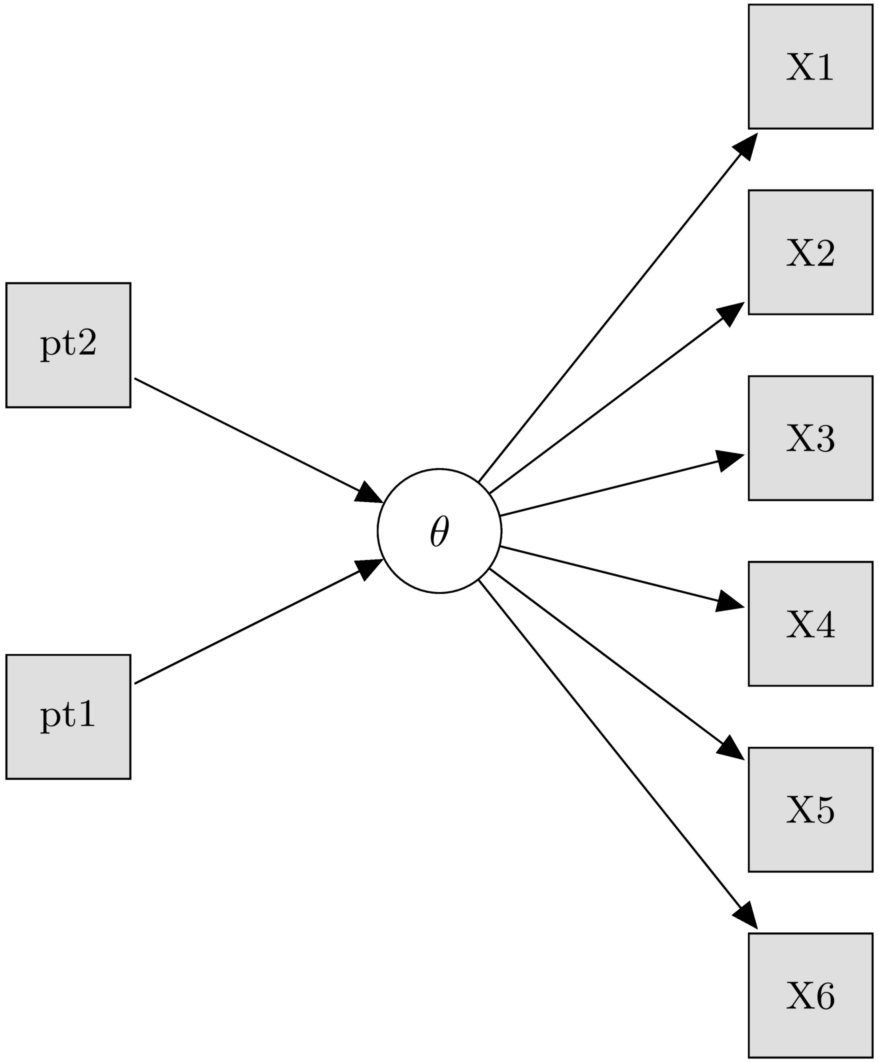
Two-step LCA analysis with covariates

- Step 1, fit the measurement model without the covariates:

▶ Code

- Step 2, fit the model with covariates fixing the measurement part:

▶ Code



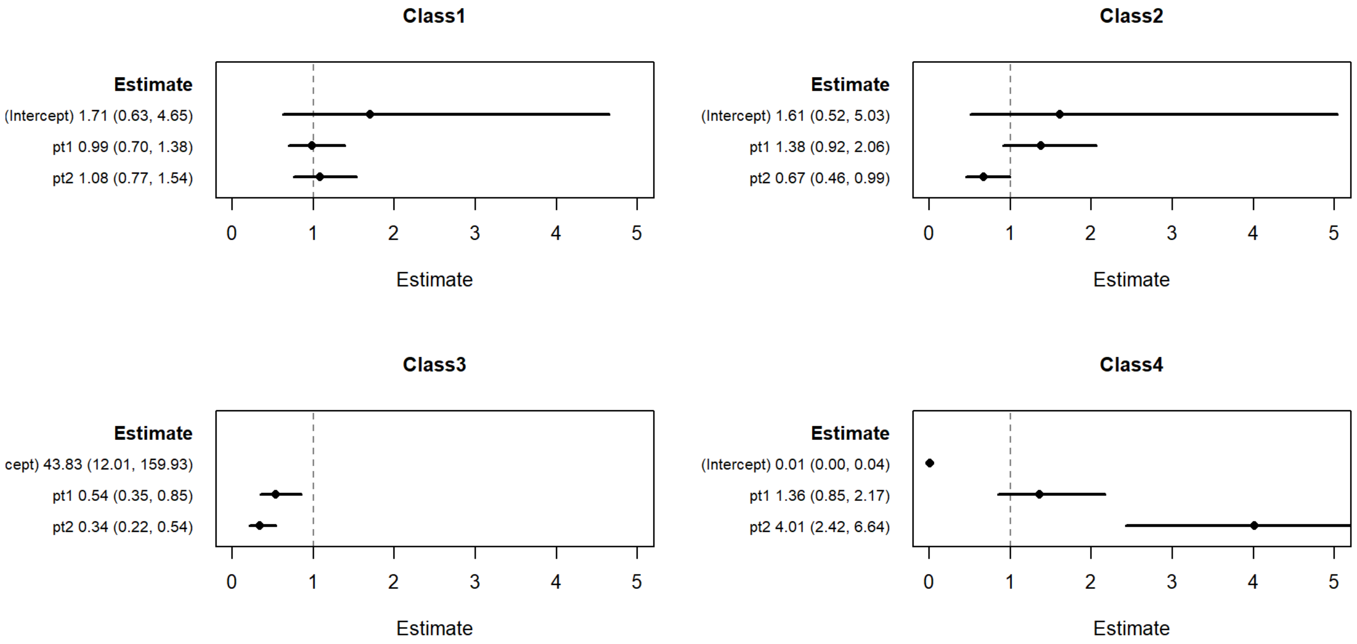
Two-step LCA analysis with covariates

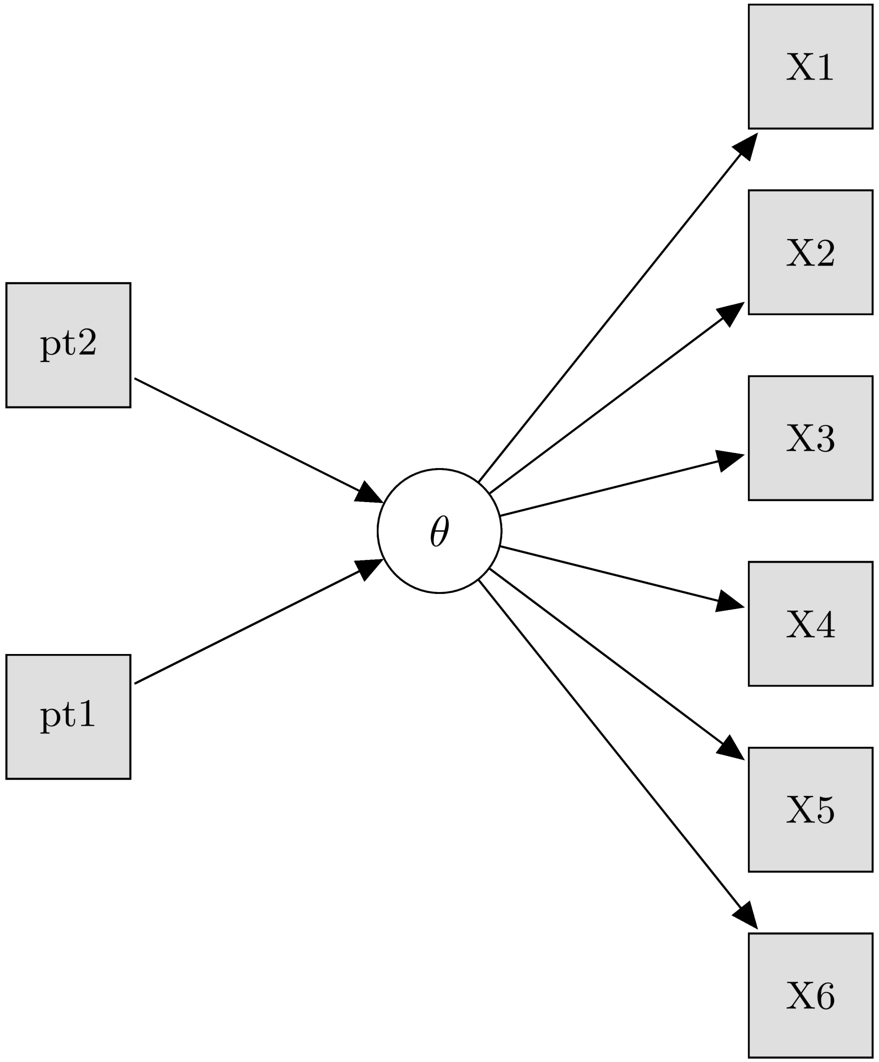
Profile output:

▶ Code

```
              Class1 Class2 Class3 Class4
(Intercept)        0 -0.058  3.246 -5.326
pt1                0  0.334 -0.598  0.320
pt2                0 -0.478 -1.155  1.309
```

▶ Code





Regularization

Introducing regularization in the model:

- ▶ Code
- ▶ Code

L-type regularization for predictors coefficients,

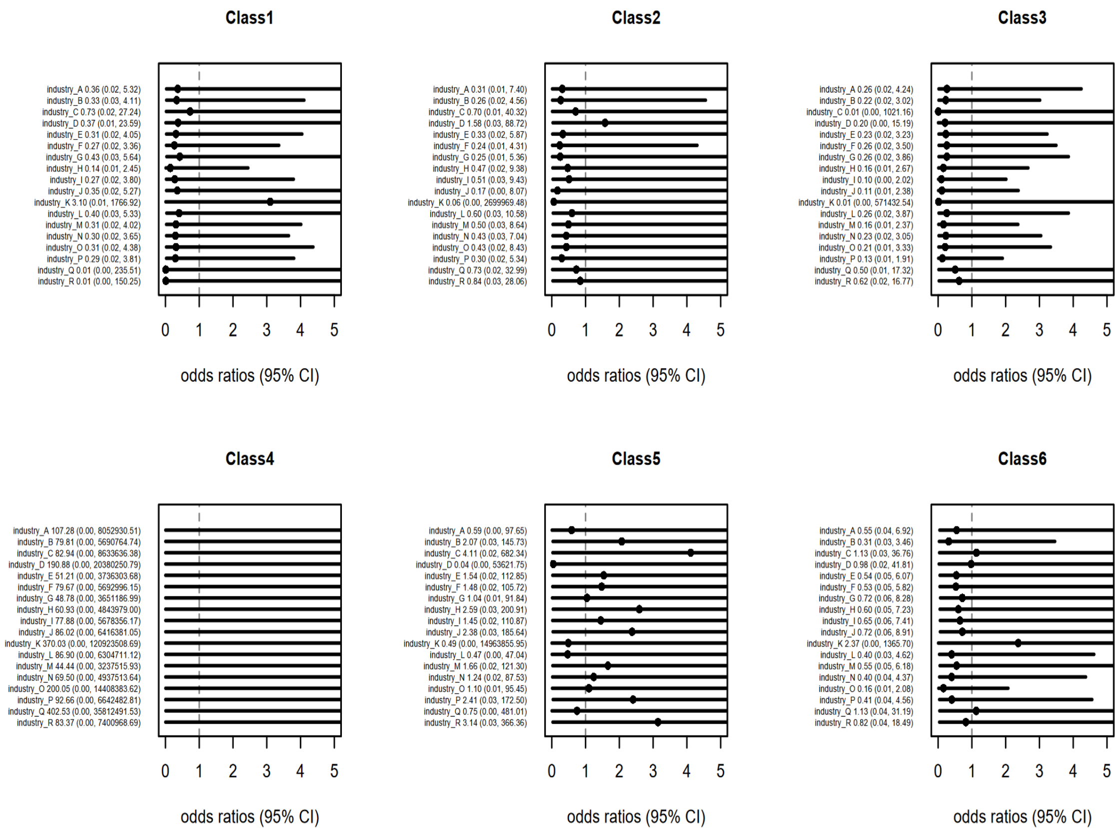
$$\lambda \ \|\beta\|_2^L,$$

or MAP estimation with a gaussian prior and precision hyperparameter α,

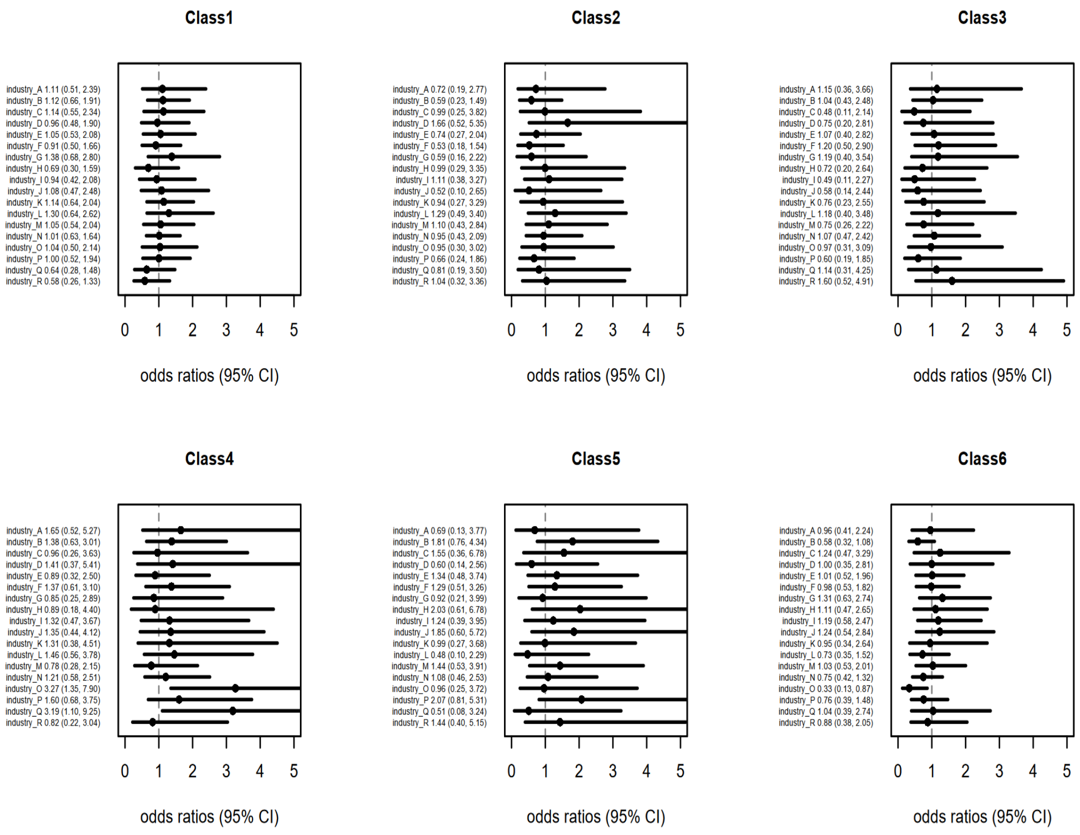
$$\beta_{std} \sim N(0, 1/\alpha).$$

Real example of regularization with MAP

Without regularization



With MAP regularization



Factor Analysis

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

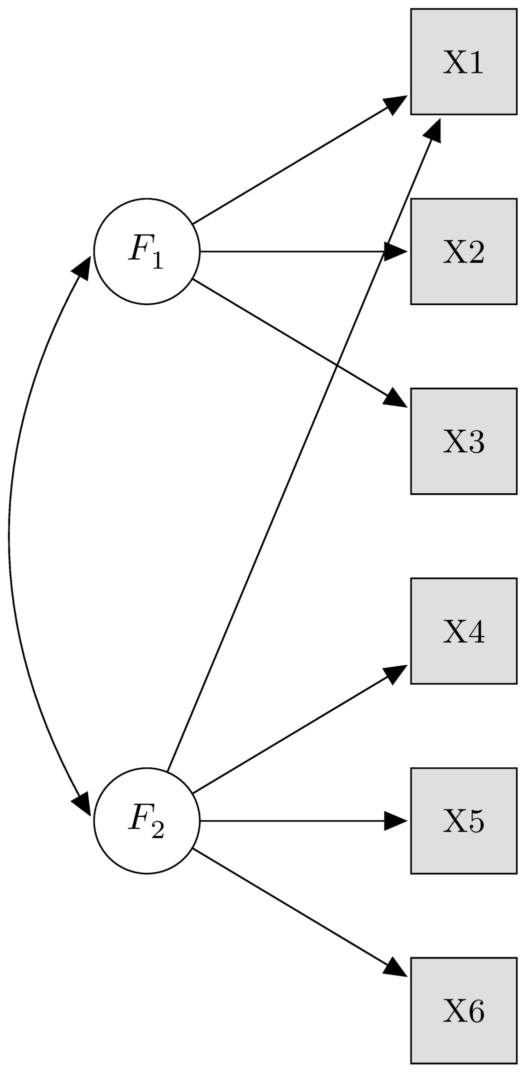
The score in item j is a weighted sum of the K latent factors:

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

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,

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

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)

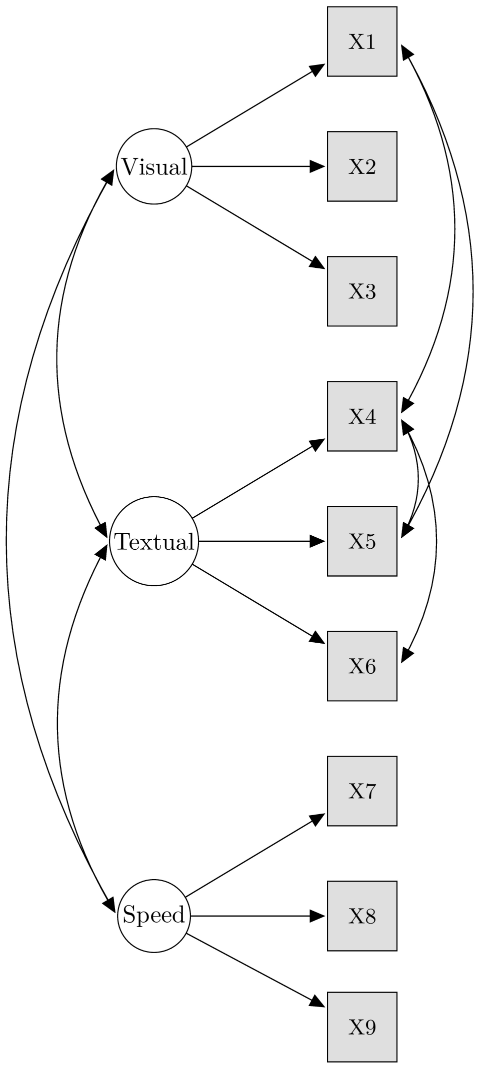
Let's force an instance where lavaan fails to converge to a proper solution.

▶ Code

```
      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
```

▶ Code

```
[1] -0.00118551
```



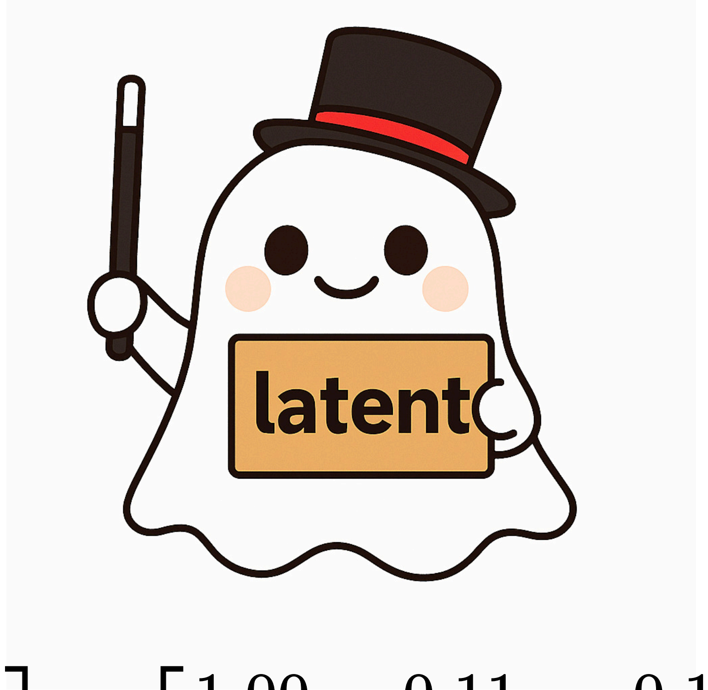
Positive-definite constraints

- Parameterize the polychoric correlation matrix as a crossproduct:

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

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

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



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

Positive-definite latent covariances

- Parameterize latent covariances as crossproducts:

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

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

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

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

Positive-definite latent covariances

▶ Code

```
      x1    x2    x3     x4    x5     x6    x7    x8    x9
x1 0.453 0.000 0.000  0.084 0.046  0.000 0.000 0.000 0.000
x2 0.000 0.807 0.000  0.000 0.000  0.000 0.000 0.000 0.000
x3 0.000 0.000 0.625  0.000 0.000  0.000 0.000 0.000 0.000
x4 0.084 0.000 0.000  0.321 0.129 -0.104 0.000 0.000 0.000
x5 0.046 0.000 0.000  0.129 0.461  0.000 0.000 0.000 0.000
x6 0.000 0.000 0.000 -0.104 0.000  0.040 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 0.461 0.000
x9 0.000 0.000 0.000  0.000 0.000  0.000 0.000 0.000 0.572
```

▶ Code

```
[1] 2.379552e-06
```

Cooking new stuff

- Standard errors for 2-step models
- Expectation-Maximization algorithm
- (Exploratory) Structural Equation Modeling
- Hidden Markov Models

Release date? Soon

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

Contact: m.j.jimenezhenriquez@vu.nl