

# Package ‘lsl’

May 8, 2026

**Type** Package

**Title** Latent Structure Learning

**Version** 0.5.6

**Author** Po-Hsien Huang [aut, cre]

**Maintainer** Po-Hsien Huang <psyphh@gmail.com>

**Description** Fits structural equation modeling via penalized likelihood.

**Depends** R (>= 3.2.0)

**License** GPL (>= 3)

**Imports** methods, ggplot2, reshape2, lavaan

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-11-08 05:30:21 UTC

## Contents

lsl . . . . .	1
lslSEM-class . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

lsl *lsl: Latent Structure Learning*

---

## Description

lsl is a package designed for conducting latent structure learning methods. In the current version, structural equation modeling via penalized likelihood can be implemented using the reference class [lslSEM](#).

---

 IslSEM-class
 

---

*A Reference Class for Learning a SEM model via penalized likelihood.*


---

## Description

A Reference Class for Learning a SEM model via penalized likelihood.

## Fields

`$data` A list of stored data.

`$model` A list of pattern matrices and starting values for model parameters.

`$structure` A list of learned structure based on the inputted data and specified model.

## Methods

`draw(type, object)` Method `draw()` is used to draw a plot for the learned structure. Argument `type` specify which type of plot should be drawn.

If `type = 'overall'`, `draw()` will give a plot for the goodness-of-fit indices across different values of  $\gamma$  and  $\delta$ . The argument `object` can be used to specify which goodness-of-fit indices should be plot. Its value can be any combination of `'dml'`, `'srmr'`, `'rmsea'`, `'mc'`, `'ghat'`, `'cfi'`, `'nnfi'`, `'bl89'`, `'rmi'`, `'aic'`, and `'bic'`. The default value is `c('dml', 'aic', 'bic')`.

If `type = 'individual'`, `draw()` will give a plot for the solution paths of parameter estimates under across different values of  $\gamma$  and  $\delta$ . The argument `object` can be used to specify parameters in which matrix should be plot. Its value can be `'lambda'`, `'psi'`, `'beta'`, `'phi'`, `'nu'`, and `'alpha'`. The default value is `'lambda'`.

`input(raw)` Method `input()` is used to input a data set for further statistical analysis.

Argument `raw` is a raw data matrix that will be used for analysis.

`learn(penalty, control, variable)` Method `learn()` is used to calculate penalized likelihood estimates based on the inputted data and specified model.

Argument `penalty` is a list for the information of penalty function and regularization parameters. Argument `penalty` includes three elements

(1) `type = c('l1', 'scad', 'mcp')`: the penalty function to be implemented. The default value is `'l1'`;

(2) `gamma = seq(0.025, .10, .025)`: the values of regularization parameter  $\gamma$  to be considered;

(3) `delta = 2.5`: the values of shape parameter  $\delta$  to be considered.

Argument `control` is a list for controlling the optimization process. Argument `control` includes two elements:

(1) `maxit = 500`, the maximum number of ECM iterations;

(2) `epsilon = 10-5`, the convergence criterion of ECM algorithm.

Argument `variable` is a vector of index of variable names to specify which variables in data should be used for analysis. The default value is `1:nrow(data$raw)`, which select all of the variables.

`specify(pattern, value, scale)` Method `specify()` is used to specify a SEM model for fitting.

Argument `pattern` is a list of pattern matrices for the six model parameter matrices in SEM. For each pattern matrix, the element can be set as 1, 0, or NA. 1 means that the corresponding parameter should be freely estimated, 0 indicates that the parameter should be fixed, and NA makes the parameter to be estimated with penalization. These pattern matrices for the six model parameter matrices include

- (1) `lambda`: a P by M pattern matrix for loading matrix;
- (2) `psi`: a P by P pattern matrix covariance matrix of measurement errors. Its default value is `diag(1, P)`. Note that the diagonal element of `psi` must be set as 1;
- (3) `beta`: a M by M pattern matrix for path coefficient matrix. Its default value is `matrix(0, M, M)`;
- (4) `phi`: a M by M pattern matrix for covariance matrix of residuals. Its default value is `diag(1, M)`. Note that the diagonal element of `phi` must be set as 1;
- (5) `nu`: a P by 1 pattern matrix for intercepts of observed variables. Its default value is `matrix(1, P, 1)`;
- (6) `alpha`: a M by 1 pattern matrix for intercepts of latent factors. Its default value is `matrix(0, M, 1)`.

Argument `value` is a list of starting value matrices for the six model parameter matrices in SEM. These starting value matrices for the six model parameter matrices include

- (1) `lambda`: a P by M starting value matrix for the loading matrix. The default value for its element is (a) 1 for the freely estimated parameters and (2) 0 for the fixed or penalized parameter.
- (2) `psi`: a P by P starting value matrix for the covariance matrix of measurement errors. Its default value is `diag(0.1, P)`.
- (3) `beta`: a M by M starting value matrix for the path coefficient matrix. Its default value is `matrix(0, M, M)`;
- (4) `phi`: a M by M starting value matrix for the covariance matrix of residuals. Its default value is `diag(1, M)`.
- (5) `nu`: a P by 1 starting value matrix for the intercepts of observed variables. Its default value is sample means of observed variables;
- (6) `alpha`: a M by 1 starting value matrix for the intercepts of latent factors. Its default value is `matrix(0, M, 1)`.

Argument `scale` is a logical indicator for specifying whether the scale of latent variable should be determined automatically. If `scale = TRUE`, the first freely estimated loading of each latent factor will be set as one for scale setting. The default value of `scale` is 1.

`summarize(type, selector, object)` Method `summarize()` is used to obtain a summary for the learned structure. Argument `type` specifies which type of summary should be made.

If `type = 'overall'`, `summarize()` will give the overall model information (including goodness-of-fit indices) under best AIC and BIC models.

If `type = 'individual'`, `summarize()` will give the parameter estimates under best AIC and BIC model.

## Examples

```
#Example: Holzinger and Swineford (1939) Data#
lambda <- matrix(NA, 9, 3)
```

```
lambda[c(1,2,3), 1] <- lambda[c(4,5,6), 2] <- lambda[c(7,8,9), 3] <- 1

rc_sem <- IsISEM()
rc_sem$input(raw = lavaan::HolzingerSwineford1939)
rc_sem$specify(pattern = list(lambda = lambda))
rc_sem$learn(penalty = list(type = "l1", gamma = seq(.05, .10, .05)), variable = 7:15)
rc_sem$draw(type = "individual", object = "lambda")
rc_sem$summarize(type = "overall")
rc_sem$summarize(type = "individual")
```

# Index

lsl, [1](#)  
lsl-package (lsl), [1](#)  
lslSEM, [1](#)  
lslSEM (lslSEM-class), [2](#)  
lslSEM-class, [2](#)