Package 'lavinteract'

November 13, 2025

Type Package

Title Post-Estimation Utilities for 'lavaan' Fitted Models

Version 0.3.4

Description Companion toolbox for structural equation models fitted with 'lavaan'. Provides post-estimation diagnostics and graphics that operate directly on a fitted object using its estimates and covariance, and refits auxiliary models when needed. The package relies on 'lavaan' (Rosseel, 2012) <doi:10.18637/jss.v048.i02>.

URL https://github.com/g-corbelli/lavinteract

BugReports https://github.com/g-corbelli/lavinteract/issues

License GPL-3

Encoding UTF-8

Imports lavaan, rlang, ggplot2, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Language en-US

NeedsCompilation no

Maintainer Giuseppe Corbelli <giuseppe.corbelli@uninettunouniversity.net>

Config/testthat/edition 3

RoxygenNote 7.3.2

Author Giuseppe Corbelli [aut, cre] (ORCID: <https://orcid.org/0000-0002-2864-3548>)

Repository CRAN

Date/Publication 2025-11-13 17:40:10 UTC

Contents

`lavinteract` *Post-Estimation Utilities for 'lavaan' Fitted Models*

Description

Companion toolbox for structural equation models fitted with 'lavaan'. Operates directly on a fitted object using its estimates and covariance. Refits auxiliary models when needed to compute estimates, diagnostics, and plots.

Details

The functions are:

- `lav_slopes`: simple slopes and interaction plots from a fitted 'lavaan' model.
- `lav_vif`: variance inflation factors for structural predictors with measurement preserved.
- `lav_cv`: repeated holdout (Monte Carlo) cross-validation of R^2 for SEM outcomes.

Note

The development of this package grew from ongoing discussions and interactions (sic) with colleagues, in particular Dr. Cataldo Giuliano Gemmano, whose steady feedback and support helped shape it.

Author(s)

Giuseppe Corbelli (<giuseppe.corbelli@uninettunouniversity.net>)

See Also

Useful links:

- https://github.com/g-corbelli/lavinteract
- Report bugs at https://github.com/g-corbelli/lavinteract/issues

`lav_cv` *Repeated holdout (Monte Carlo) cross-validation of R^2 for structural equation models ('lavaan' objects)*

Description

Estimate out-of-sample predictive performance for structural relations in a fitted 'lavaan' model using repeated holdout (Monte Carlo cross-validation, leave-group-out CV). At each repetition, the model is refitted on a random training subset and evaluated on a disjoint test subset.

Usage

```
lav_cv(
  fit,
  data = NULL,
  times = "auto",
  train_prop = 0.8,
  seed = 42L,
  quiet = TRUE,
  digits = 3L,
  plot = TRUE,
  tol = 0.001,
  window = 50L,
  max_times = 3000L,
  min_r2_for_pct = 0.05
)

## S3 method for class 'lav_cv'
print(x, digits = x$digits %||% 3L, ...)

## S3 method for class 'lav_cv'
summary(object, ...)
```

Arguments

fit	A fitted 'lavaan' object (required).
data	The data frame used to fit the model; if NULL, it is extracted from 'fit' when available (default: NULL).
times	Integer indicating the number of random splits, or "auto" for stabilization-based early stopping (default: "auto").
train_prop	Numeric in (0,1). Proportion of cases in the training split for each repetition (default: 0.8).
seed	Integer. Random seed for reproducibility of the splits (default: 42).
quiet	Logical. Suppress 'lavaan' refit messages when TRUE (default: TRUE).
digits	Integer. Number of digits to print in summaries (default: 3).
plot	Logical. Show convergence plots of the running mean R^2 per outcome (default: TRUE).
tol	Numeric. Tolerance for the auto-stop rule on the running mean (default: 0.001).
window	Integer. Trailing window size (number of successful splits) used by the auto-stop rule (default: 50).
max_times	Integer. Maximum number of splits when times = "auto" (default: 3000).
min_r2_for_pct	Numeric in (0,1). Minimum in-sample R^2 required to compute percent drop; below this, %_drop is set to NA (default: 0.05).
x	A 'lav_cv' object.
...	Additional arguments; unused.
object	A 'lav_cv' object.

Details

For observed outcomes, R^2 is computed by comparing test-set observed values with predictions obtained by applying the training-set structural coefficients to the test-set predictors.

For latent outcomes, the outcome is not directly observed in the test set. Factor scores for the outcome are first computed in the test set using the measurement model learned on the training set; these scores serve as the outcome values. Predictions are then formed by applying the training-set structural coefficients to the test-set predictors (including factor scores for any latent predictors). R^2 is computed by comparing the test-set factor scores of the outcome with these predicted scores.

The in-sample baseline R^2 is computed on the full dataset using the same metric as in cross-validation: observed outcomes use observed-versus-predicted R^2; latent outcomes use score-versus-predicted-score R^2.

By default, repetitions continue until the running mean R^2 for each outcome stabilizes within a specified tolerance over a trailing window of successful splits, or until a maximum number of splits is reached.

The summary table reports the in-sample baseline R^2, the median cross-validated R^2, its standard deviation, and the percent drop (baseline vs. median CV) with heuristic threshold markers. The percent drop is suppressed when the in-sample R^2 is very small.

Value

A list with class 'lav_cv' and elements:

`table` Data frame with columns: `outcome`, `type` ("observed" or "latent"), `r2_in`, `r2_cv_mean`, `r2_cv_median`, `r2_cv_sd`, `drop_mean_pct`, `drop_med_pct`, `splits_used`.

`split_matrix` Matrix of split-wise test-set R^2 values (rows = splits, columns = outcomes).

`times` Character or integer indicating the number of splits used (e.g., `"auto(534)"` or `500`).

`train_prop` Numeric. Training proportion used in each split.

`N` Integer. Number of rows in the input data.

`seed` Integer. Random seed used to generate the splits.

`tol` Numeric. Tolerance used by the auto-stop rule.

`window` Integer. Trailing window size for the auto-stop rule.

`min_r2_for_pct` Numeric. Minimum in-sample R^2 required to compute percent drop.

`call` `match.call()` of the function call.

`digits` Integer. Default number of digits for printing.

References

Cudeck, R., & Browne, M. W. (1983). Cross-Validation Of Covariance Structures. Multivariate Behavioral Research, 18(2), 147-167. doi:10.1207/s15327906mbr1802_2

Hastie, T., Friedman, J., & Tibshirani, R. (2001). The Elements of Statistical Learning. In Springer Series in Statistics. Springer New York. doi:10.1007/9780387216065

Kvalseth, T. O. (1985). Cautionary Note about R2. The American Statistician, 39(4), 279-285. doi:10.1080/00031305.1985.10479448

Shmueli, G. (2010). To Explain or to Predict? Statistical Science, 25(3). doi:10.1214/10sts330

Yarkoni, T., & Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. Perspectives on Psychological Science, 12(6), 1100-1122. doi:10.1177/1745691617693393

See Also

sem, lavPredict, inspect

Examples

```
library("lavaan")
model <- "
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8

dem60 ~ ind60
dem65 ~ ind60 + dem60

y1 ~~ y5
y2 ~~ y6
"
fit <- lavaan::sem(
model = model,
data = lavaan::PoliticalDemocracy,
std.lv = TRUE,
estimator = "MLR",
meanstructure = TRUE)

result <- lav_cv(
fit = fit,
data = lavaan::PoliticalDemocracy,
times = 5)
print(result)
```

---

lav_slopes *Simple slopes and interaction plots for fitted 'lavaan' models*

Description

Computes conditional (simple) slopes of a focal predictor across values of a moderator from a fitted 'lavaan' model that includes their explicit product term. Plots predicted lines with Wald confidence ribbons, and print an APA-style test of the interaction for easy reporting and interpretation, plus a simple-slopes table.

Usage

```
lav_slopes(
  fit,
  outcome,
  pred,
  modx,
  interaction,
  data = NULL,
  modx.values = NULL,
  modx.labels = NULL,
  pred.range = NULL,
  conf.level = 0.95,
  x.label = NULL,
  y.label = NULL,
  legend.title = NULL,
  colors = NULL,
  line.size = 0.80,
  alpha = 0.20,
  table = TRUE,
  digits = 2,
  modx_n_unique_cutoff = 4L,
  return_data = FALSE
)

## S3 method for class 'lav_slopes'
print(x, ...)

## S3 method for class 'lav_slopes'
summary(object, ...)
```

Arguments

`fit`	A fitted 'lavaan' object that includes the product term (required).
`outcome`	Character. Name of the dependent variable in `fit` (required).
`pred`	Character. Name of the focal predictor whose simple slopes are probed (required).
`modx`	Character. Name of the moderator (required).
`interaction`	Character. Name of the product term in `fit` (e.g., `"X_Z"`) (required).
`data`	`data.frame`. Raw data. If `NULL`, the function tries to pull data from `fit` via `lavInspect`.
`modx.values`	Numeric or character vector. Values or levels of the moderator at which to compute slopes; derived automatically when `NULL`.
`modx.labels`	Character vector. Legend/table labels for `modx.values` (default: the character form of `modx.values`).
`pred.range`	Numeric length-2. Range `c(min, max)` for the x-axis; uses observed range in `data` when available, else `c(-2, 2)`.

conf.level	Numeric in (0,1). Confidence level for CIs and ribbons (default: 0.95).
x.label	Character. X-axis label (default: pred).
y.label	Character. Y-axis label (default: outcome).
legend.title	Character. Legend title; if NULL, the legend shows only levels (default: NULL).
colors	Character vector. Colors for lines and ribbons; named vector recommended with names matching modx.labels (default: Okabe-Ito palette).
line.size	Numeric > 0. Line width (default: 0.80).
alpha	Numeric in (0,1). Ribbon opacity (default 0.20).
table	Logical. Print APA-style interaction test and simple-slopes table (default: TRUE).
digits	Integer >= 0. Decimal digits in printed output (default: 2).
modx_n_unique_cutoff	Integer >= 1. Threshold for treating a numeric moderator as continuous and using mean ± SD (default: 4).
return_data	Logical. If TRUE, include the plotting data.frame in the returned list (default: FALSE).
x	A 'lav_slopes' object.
...	Additional arguments; unused.
object	A 'lav_slopes' object.

Details

The model should include a main effect for the predictor, a main effect for the moderator, and their product term. The simple slope of the predictor at a given moderator value combines the predictor main effect with the interaction term. The moderator can be continuous or categorical. Standard errors use the delta method with the model covariance matrix of the estimates.

Value

A list with elements:

plot ggplot object with lines and confidence ribbons.

slope_table Data frame with moderator levels, simple slopes, SE, z, and CI.

plot_data Only when return_data = TRUE: data used to build the plot.

Notes

Estimates are unstandardized; a standardized beta for the interaction is also reported for reference. Wald tests assume large-sample normality of estimates.

Examples

```
set.seed(42)
X <- rnorm(100); Z <- rnorm(100); X_Z <- X*Z
Y <- 0.6*X + 0.6*Z + 0.3*X_Z + rnorm(100, sd = 0.7)
dataset <- data.frame(Y, X, Z, X_Z)
fit <- lavaan::sem("Y ~ X + Z + X_Z", data = dataset)
```

```
lav_slopes(
fit = fit,
data = dataset,
outcome = "Y",
pred = "X",
modx = "Z",
interaction = "X_Z")
```

---

lav_vif *Variance Inflation Factors for 'lavaan' Structural Predictors*

Description

Compute VIF for each predictor that appears in structural regressions with two or more predictors, refitting the necessary sub-models so that latent predictors are handled at the latent level (i.e., with their original measurement models). It returns also the R^2 of each eligible endogenous variable from the original fit for context.

Usage

```
lav_vif(
  fit,
  data = NULL,
  quiet = TRUE
)

## S3 method for class 'lav_vif'
print(x, digits = 3, cutoff = c(5, 10), ...)

## S3 method for class 'lav_vif'
summary(object, ...)
```

Arguments

fit	A fitted lavaan object.
data	Optional. The data frame used to fit fit. If NULL, the function attempts to extract the data from fit via lavInspect(fit, "data") then "data.original".
quiet	Logical. If TRUE suppresses lavaan refit messages.
x	A 'lav_vif' object.
digits	Integer number of digits to print.
cutoff	Numeric length-2 thresholds used to flag VIF values.
...	Passed to 'print.lav_vif()' (e.g., 'digits', 'cutoff').
object	A 'lav_vif' object.

Details

Each auxiliary refitted model:

- includes the original measurement model for any latent predictors;
- includes any residual covariances among those indicators that were specified in the original model;
- regresses the focal predictor on the remaining predictors at the latent level when applicable.

VIF_i = 1 / (1 - R^2_i) generalizes VIF to SEM while respecting measurement models.

The function reuses the estimator, missing-data handling, and several options from `fit`.

Value

A list with:

- `vif_table`: data.frame with columns `outcome`, `predictor`, `group`, `r2_predictor`, `vif`, `k_predictors`.
- `outcome_r2`: data.frame with R^2 per eligible endogenous `outcome` and `group` from the original `fit`.

Examples

```
set.seed(42)
x1 <- rnorm(100); x2 <- 0.85*x1 + rnorm(100, sd = sqrt(1 - 0.85^2)); x3 <- rnorm(100)
y  <- 0.5*x1 + 0.3*x2 + 0.1*x3 + rnorm(100, sd = 0.7)
dataset <- data.frame(y, x1, x2, x3)
fit <- lavaan::sem("y ~ x1 + x2 + x3", data = dataset)
lav_vif(
fit = fit,
data = dataset)
```

Index