

Package ‘semlrtp’

July 19, 2026

Title Likelihood Ratio Test P-Values for Structural Equation Models

Version 0.1.2

Description Computes likelihood ratio test (LRT) p-values
for free parameters in a structural equation model.
Currently supports models fitted by the 'lavaan' package by
Rosseel (2012) <[doi:10.18637/jss.v048.i02](https://doi.org/10.18637/jss.v048.i02)>.

License GPL (>= 3)

Encoding UTF-8

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

Imports lavaan (>= 0.6-18), utils, parallel, pbapply

VignetteBuilder knitr

URL <https://sfcheung.github.io/semlrtp/>

BugReports <https://github.com/sfcheung/semlrtp/issues>

Config/Needs/website rmarkdown

Depends R (>= 4.0.0)

LazyData true

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Shu Fai Cheung [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9871-9448>>),
Mark Hok Chio Lai [aut] (ORCID:
<<https://orcid.org/0000-0002-9196-7406>>)

Maintainer Shu Fai Cheung <shufai.cheung@gmail.com>

Repository CRAN

Date/Publication 2026-07-19 11:00:02 UTC

Contents

data_sem16	2
fix_to_zero	3
lrt	5
lrtp	7
print.lrt	9
print.lrtp	10
Index	13

data_sem16	<i>Sample Dataset For Test (16 Items and 2 Groups)</i>
------------	--

Description

A 16-variable dataset with 336 cases.

Usage

data_sem16

Format

A data frame with 336 rows and 16 variables:

- x1** Indicator. Numeric.
- x2** Indicator. Numeric.
- x3** Indicator. Numeric.
- x4** Indicator. Numeric.
- x5** Indicator. Numeric.
- x6** Indicator. Numeric.
- x7** Indicator. Numeric.
- x8** Indicator. Numeric.
- x9** Indicator. Numeric.
- x10** Indicator. Numeric.
- x11** Indicator. Numeric.
- x12** Indicator. Numeric.
- x13** Indicator. Numeric.
- x14** Indicator. Numeric.
- x15** Indicator. Numeric.
- x16** Indicator. Numeric.
- group** Group with two values, "alpha" and "gamma". Character.

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
f1 =~ x1 + x2 + x3 + x4
f2 =~ x5 + x6 + x7 + x8
f3 =~ x9 + x10 + x11 + x12
f4 =~ x13 + x14 + x15 + x16
f3 ~ f2 + f1
f4 ~ f3
"
fit <- sem(mod, data_sem16)
summary(fit)
```

fix_to_zero

Fix a Free Parameter To Zero And Fit a Model Again

Description

It fixes a designated free parameter in a lavaan object to zero and refit the model.

Usage

```
fix_to_zero(fit, par_id, store_fit = FALSE, se_keep_bootstrap = FALSE)
```

Arguments

fit	A lavaan-class object.
par_id	An integer. The row number of the free parameter in the parameter table of fit to be fixed.
store_fit	Logical. If TRUE, fit will be stored in the output. Default is FALSE.
se_keep_bootstrap	Logical. If TRUE and fit used bootstrapping standard error (with se = "bootstrap"), bootstrapping will also be use in fitting the restricted model. If FALSE, the default, then se will be set to "standard" if it is "bootstrap" in fit, to speed up the computation.

Details

It modifies the parameter table of a lavaan-class object and then fits the model again.

Users should usually call `lrtp()` directly instead of calling this function. It is exported for developers.

Value

A `fix_to_zero`-class object, which is a list with these elements:

- `fit0` is the lavaan output of the refitted object. NA if the fit failed for some reasons. To be considered an acceptable solution, the optimization must converge, the solution passes lavaan's post check, the variance-covariance matrix of estimates successfully computed, and the increase in the model degree of freedom equal to the expected change.
- `fit1` is the original lavaan output if `store_fit` is TRUE. It is NULL if `store_fit` is FALSE, the default.
- `par_id` is the row number of the designated free parameter in the parameter table.
- `call` is the original call to this function.
- `ptable0` is the parameter table with the designated parameter fixed to zero. It can be used for diagnostic purpose if the fit failed.
- `fit0_error` is the error message in refitting the model (`ptable0`), if any. If no error, it is NA.
- `vcov_ok` is TRUE if the variance-covariance matrix of the estimates can be computed without error nor warning. FALSE otherwise.
- `vcov_msg` is the message generated when using `lavaan::lavInspect()` to get the variance-covariance matrix of the parameter estimates of the refitted model. If TRUE, then no error nor warning. Can be used for diagnostic purposes.
- `converged`: Whether refitting the modified model (`ptable0`) converged.
- `post_check_passed`: Whether the solution of the modified model (`ptable0`) passed lavaan's post check.
- `post_check_msg`: If the solution failed lavaan's post check, it stores the warning message. If the solution passes the check, it is NA.
- `fit_not_ok`: If the fit failed for some reasons, the fit object, if available, is stored in this element rather than in `fit0`. such that the fit object can be retrieved for diagnostic purposed if necessary.
- `df_diff_one`: Whether the difference in model degrees of freedom between the modified model and the original model is one. If a variance is fitted to zero, related covariance(s) is/are also fitted to zero and so the difference in model degrees of freedom can be legitimately greater than one.
- `se_force_standard`: Whether se was forced to be "standard" even if it is "bootstrap" in fit. If FALSE, then either se is not "bootstrap" in fit or it was not changed in fitting the restricted model.

Author(s)

Shu Fai Cheung <https://orcid.org/0000-0002-9871-9448>

See Also

[lrrtp\(\)](#)

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
"

fit <- sem(mod, data_sem16)
# Fix the factor covariance to zero
out <- fix_to_zero(fit, par_id = 15)
summary(out$fit0)
```

lrt

Fix a Free Parameter to Zero and Do an LR Test

Description

Fix the designated free parameter to zero and do a likelihood ratio test.

Usage

```
lrt(
  fit,
  par_id,
  store_fit = FALSE,
  group = NULL,
  se_keep_bootstrap = FALSE,
  LRT_method = "default",
  scaled_shifted = TRUE,
  fallback_method = "satorra.2000"
)
```

Arguments

<code>fit</code>	A lavaan-class object.
<code>par_id</code>	It can be an integer. or a string. If it is an integer, it should be the row number of the free parameter in the parameter table of <code>fit</code> to be fixed to zero. If it is a string, it must be a valid lavaan model syntax for a parameter, or the label of a labelled parameter.
<code>store_fit</code>	Logical. If TRUE, <code>fit</code> will be stored in the output. Default is FALSE.
<code>group</code>	If a model syntax is used in <code>par_id</code> and the model is a multigroup model, this should be either the group label or the group number of the parameter.

se_keep_bootstrap	Logical. If TRUE and fit used bootstrapping standard error (with se = "bootstrap"), bootstrapping will also be use in fitting the restricted model. If FALSE, the default, then se will be set to "standard" if it is "bootstrap" in fit, to speed up the computation.
LRT_method	String. Passed to the method argument of <code>lavaan::lavTestLRT()</code> . Default is "default", and let <code>lavaan::lavTestLRT()</code> decide the method based on fit.
scaled.shifted	Logical. Used when the method used in <code>lavaan::lavTestLRT()</code> is "satorra.2000". Default is TRUE and a scaled and shifted test statistic is used, the same default of <code>lavaan::lavTestLRT()</code> .
fallback_method	The default method of <code>lavaan::lavTestLRT()</code> , "satorra.bentler.2001", may sometimes fail. If failed, this function will call <code>lavaan::lavTestLRT()</code> again using fallback_method. which is "satorra.2000" by default.

Details

It fixes the designated free parameter in a lavaan output, refit the model, and do a likelihood ratio test comparing this model with the original model.

The model to be fixed is generated by `fix_to_zero()`.

If the parameter to be fixed is a variance, related covariance(s), if any, will also be fixed to zero.

Users should usually call `lrtp()` directly instead of calling this function. It is exported for developers.

Value

A lrt-class object, which is a list with the following elements:

- lrt: The output of `lavaan::lavTestLRT()`. If there is an error message or warning, it is set to NA.
- par_id: The row number of the designated free parameter.
- par_label: The label of the designated free parameter, generated by `lavaan::lav_partable_labels()`.
- fit1: The original lavaan output, if store_fit is TRUE. NA if store_fit is FALSE, the default.
- fix_to_zero: The output of `fix_to_zero()`.
- call: The call to this function.
- lrt_status: Integer. If 0, then there is no error nor warning in the likelihood ratio test and `lavaan::lavTestLRT()` returns a table (data.frame) of the test. If -1, then something is wrong, e.g., an error or warning occurred when doing the likelihood ratio test.
- lrt_msg: If something went wrong when doing the likelihood ratio test, this is the error or warning message when calling `lavaan::lavTestLRT()`. If no error nor warning, this is NA.

Author(s)

Shu Fai Cheung <https://orcid.org/0000-0002-9871-9448>

See Also

`print.lrt()` for its print-method, and `lrtp()` for the main function.

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
"

fit <- sem(mod, data_sem16)
# Fix the factor covariance to zero
out <- lrt(fit, par_id = 15)
out$lrt
parameterEstimates(fit)[15, ]
parameterEstimates(out$fix_to_zero$fit0)[15, ]

# Can use model syntax for par_id

out <- lrt(fit, par_id = "f1 =~ x3")
out$lrt
```

lrtp

Likelihood Ratio Test P-Values

Description

Compute the likelihood ratio test (LRT) p -values for free parameters in a lavaan output.

Usage

```
lrtp(
  fit,
  op = c("~", "~~"),
  no_variances = TRUE,
  no_error_variances = TRUE,
  no_error_covariances = TRUE,
  se_keep_bootstrap = FALSE,
  LRT_method = "default",
  scaled_shifted = TRUE,
  fallback_method = "satorra.2000",
  progress = TRUE,
  parallel = FALSE,
  ncores = parallel::detectCores(logical = FALSE) - 1,
  load_balancing = TRUE,
  ...
)
```

Arguments

<code>fit</code>	A lavaan-class object.
<code>op</code>	A character vector of lavaan model syntax operators. Free parameters of these operators will be included, unless excluded by other arguments. Default is <code>c("~", "~~")</code> .
<code>no_variances</code>	Logical. If TRUE, the default, then all free variances are excluded. (Error variances are handled by <code>no_error_variances</code> .)
<code>no_error_variances</code>	Logical. If TRUE, the default, then all free error variances are excluded.
<code>no_error_covariances</code>	Logical. If TRUE, the default, then all free error covariances are excluded.
<code>se_keep_bootstrap</code>	Logical. If TRUE and <code>fit</code> used bootstrapping standard error (with <code>se = "bootstrap"</code>), bootstrapping will also be use in fitting the restricted model. If FALSE, the default, then <code>se</code> will be set to "standard" if it is "bootstrap" in <code>fit</code> , to speed up the computation.
<code>LRT_method</code>	String. Passed to the method argument of <code>lavaan::lavTestLRT()</code> . Default is "default", and let <code>lavaan::lavTestLRT()</code> decide the method based on <code>fit</code> .
<code>scaled.shifted</code>	Logical. Used when the method used in <code>lavaan::lavTestLRT()</code> is "satorra.2000". Default is TRUE and a scaled and shifted test statistic is used, the same default of <code>lavaan::lavTestLRT()</code> .
<code>fallback_method</code>	The default method of <code>lavaan::lavTestLRT()</code> , "satorra.bentler.2001", may sometimes fail. If failed, this function will call <code>lavaan::lavTestLRT()</code> again using <code>fallback_method</code> . which is "satorra.2000" by default.
<code>progress</code>	Logical. If TRUE, the default, a progress bar will be displayed to show the progress (using the pbapply package).
<code>parallel</code>	Logical. If TRUE, parallel processing will be used to compute the LRT <i>p</i> -values for selected parameters. Default is FALSE. Set it to TRUE if the number of selected parameters is large.
<code>ncores</code>	Integer. The number of CPU cores to use if <code>parallel</code> is TRUE. Default is the number of physical cores (determined by <code>parallel::detectCores()</code>) minus 1.
<code>load_balancing</code>	Logical. If TRUE, the default, and <code>parallel</code> is TRUE, then load balancing will be used. May shorten the total processing time if the time to compute LRT <i>p</i> -values vary a lot across parameters.
<code>...</code>	Optional arguments to be passed to <code>lavaan::parameterEstimates()</code> .

Details

It finds free parameters in a lavaan-class object, computes the likelihood ratio test (LRT) *p*-value for each of them when fixed to zero, and returns a parameter estimates table with the LRT *p*-values included.

By default, it only computes LRT *p*-values for regression paths and covariances, except for error covariances. This default can be overridden using arguments such as `op`, `no_variances`, `no_error_variances`, and `no_error_covariances`.

Technical Details:

It first identify the parameters to be processed, and then call `lrt()` on each of them. Please refer to https://sfcheung.github.io/semlrtp/articles/internal_workflow.html for the internal workflow.

Value

An `lrt`-class object, which is a data-frame-like object similar to the output of `lavaan::parameterEstimates()`, with a column `LRTp` for the LRT p -values, as well as other columns such as the chi-square difference in the test. it has a print method, `print.lrt()`.

Author(s)

Shu Fai Cheung <https://orcid.org/0000-0002-9871-9448>

See Also

`print.lrt()`

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
f3 =~ x7 + x8 + x9
f4 =~ x10 + x11 + x12
f2 ~~ f1
f3 ~ f1 + f2
f4 ~ f3
"

fit <- sem(mod, data_sem16)
lrtp(fit)
lrtp(fit, op = "~")
```

```
print.lrt
```

Print an 'lrt'- Class Object

Description

Print the content of an `lrt`-class object.

Usage

```
## S3 method for class 'lrt'
print(x, digits = 3, ...)
```

Arguments

<code>x</code>	An lrt-class object.
<code>digits</code>	Integer. The number of decimal places to print. Default is 3.
<code>...</code>	Optional arguments. Not used.

Details

It is the print method for the output of `lrt()`.

Value

`x` is returned invisibly. Called for its side effect.

Author(s)

Shu Fai Cheung <https://orcid.org/0000-0002-9871-9448>

See Also

`lrtp()`

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
"
fit <- sem(mod, data_sem16)

out <- lrt(fit, par_id = "f1 ~~ f2")
out
```

`print.lrtp`*Print an 'lrtp'- Class Object*

Description

Print the content of an lrt-class object.

Usage

```
## S3 method for class 'lrt'
print(
  x,
  digits = 3,
  lrt_only = TRUE,
  wald_stats = FALSE,
  output = c("text", "data.frame", "table"),
  ...
)
```

Arguments

x	An lrt-class object.
digits	Integer. The number of decimal places to print. Default is 3.
lrt_only	Logical. If TRUE, the default, only parameters with LRT p -values will be printed.
wald_stats	Logical. If TRUE, the usual Wald statistics (e.g., z statistics, p -values, CIs) are printed. FALSE by default, assuming that users prefer using LRT statistics when using <code>lrt()</code> .
output	The format of the printout. If "text", then the style in the <code>summary()</code> of the lavaan-class object is used. If "data.frame" or "table", then the data frame format of <code>lavaan::parameterEstimates()</code> is used.
...	Optional arguments. Not used.

Details

The print method for the output of `lrt()`.

Additional diagnostic information will be printed if one or more likelihood tests encounter some errors or warnings.

Value

x is returned invisibly. Called for its side effect.

Author(s)

Shu Fai Cheung <https://orcid.org/0000-0002-9871-9448>

See Also

`lrt()`

Examples

```
library(lavaan)
data(data_sem16)
mod <-
"
```

```
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
"
fit <- sem(mod, data_sem16)

out <- lrt(fit)
out
print(out, lrt_only = FALSE)
```

Index

* **datasets**

data_sem16, [2](#)

data_sem16, [2](#)

fix_to_zero, [3](#)

fix_to_zero(), [6](#)

lavaan::lav_partable_labels(), [6](#)

lavaan::lavInspect(), [4](#)

lavaan::lavTestLRT(), [6](#), [8](#)

lavaan::parameterEstimates(), [8](#), [9](#), [11](#)

lrt, [5](#)

lrt(), [9](#), [10](#)

lrtp, [7](#)

lrtp(), [3](#), [4](#), [6](#), [7](#), [10](#), [11](#)

parallel::detectCores(), [8](#)

print.lrt, [9](#)

print.lrt(), [7](#)

print.lrtp, [10](#)

print.lrtp(), [9](#)

summary(), [11](#)