

Package ‘seminrExtras’

January 29, 2026

Type Package

Title Conduct Additional Modeling and Analysis for 'seminr'

Version 0.9.0

Description Supplemental functions for estimating and analysing structural equation models including Cross Validated Prediction and Testing (CVPAT, Lien-gaard et al., 2021 <[doi:10.1111/dec.12445](https://doi.org/10.1111/dec.12445)>).

Imports seminr (>= 2.4.0), stats

License GPL-3

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown, psych, learnr, paran

Config/testthat/edition 3

URL <https://github.com/sem-in-r/seminr>

BugReports <https://github.com/sem-in-r/seminr/issues>

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Nicholas Patrick Danks [aut, cre],
Soumya Ray [aut, ths]

Maintainer Nicholas Patrick Danks <nicholasdanks@hotmail.com>

Repository CRAN

Date/Publication 2026-01-29 14:30:02 UTC

Contents

assess_cvpat	2
assess_cvpat_compare	3
congruence_test	6
Index	8

assess_cvpat

*SEMinR function to compare CVPAT loss of two models***Description**

‘assess_cvpat’ conducts a single model CVPAT assessment against item average and linear model prediction benchmarks.

Usage

```
assess_cvpat(
  seminr_model,
  testtype = "two.sided",
  nboot = 2000,
  seed = 123,
  technique = predict_DA,
  noFolds = NULL,
  reps = NULL,
  cores = NULL
)
```

Arguments

seminr_model	The SEMinR model for CVPAT analysis
testtype	Either "two.sided" (default) or "greater".
nboot	The number of bootstrap subsamples to execute (defaults to 2000).
seed	The seed for reproducibility (defaults to 123).
technique	predict_EA or predict_DA (default).
noFolds	Number of folds for k-fold cross validation.
reps	Number of repetitions for cross validation.
cores	Number of cores for parallelization.

Value

A matrix of the estimated loss and results of significance testing.

References

Sharma, P. N., Liengaard, B. D., Hair, J. F., Sarstedt, M., & Ringle, C. M. (2022). Predictive model assessment and selection in composite-based modeling using PLS-SEM: extensions and guidelines for using CVPAT. *European journal of marketing*, 57(6), 1662-1677.

Liengaard, B. D., Sharma, P. N., Hult, G. T. M., Jensen, M. B., Sarstedt, M., Hair, J. F., & Ringle, C. M. (2021). Prediction: coveted, yet forsaken? Introducing a cross-validated predictive ability test in partial least squares path modeling. *Decision Sciences*, 52(2), 362-392.

Examples

```
# Load libraries
library(seminr)
library(seminrExtras)

# Create measurement model ----
corp_rep_mm_ext <- constructs(
  composite("QUAL", multi_items("qual_", 1:8), weights = mode_B),
  composite("PERF", multi_items("perf_", 1:5), weights = mode_B),
  composite("CSOR", multi_items("csor_", 1:5), weights = mode_B),
  composite("ATTR", multi_items("attr_", 1:3), weights = mode_B),
  composite("COMP", multi_items("comp_", 1:3)),
  composite("LIKE", multi_items("like_", 1:3))
)

# Create structural model ----
corp_rep_sm_ext <- relationships(
  paths(from = c("QUAL", "PERF", "CSOR", "ATTR"), to = c("COMP", "LIKE"))
)

# Estimate the model ----
corp_rep_pls_model_ext <- estimate_pls(
  data = corp_rep_data,
  measurement_model = corp_rep_mm_ext,
  structural_model = corp_rep_sm_ext,
  missing = mean_replacement,
  missing_value = "-99")

# Assess the base model ----
assess_cvpat(seminr_model = corp_rep_pls_model_ext,
  testtype = "two.sided",
  nboot = 20,
  seed = 123,
  technique = predict_DA,
  noFolds = 5,
  reps = 1,
  cores = 1)
```

assess_cvpat_compare *SEMinR function to compare CVPAT loss of two models*

Description

‘assess_cvpat_compare’ conducts a CVPAT significance test of loss between two models.

Usage

```
assess_cvpat_compare(
```

```

    established_model,
    alternative_model,
    testtype = "two.sided",
    nboot = 2000,
    seed = 123,
    technique = predict_DA,
    noFolds = NULL,
    reps = NULL,
    cores = NULL
  )

```

Arguments

<code>established_model</code>	The base semnr model for CVPAT comparison.
<code>alternative_model</code>	The alternate semnr model for CVPAT comparison.
<code>testtype</code>	Either "two.sided" (default) or "greater".
<code>nboot</code>	The number of bootstrap subsamples to execute (defaults to 2000).
<code>seed</code>	The seed for reproducibility (defaults to 123).
<code>technique</code>	predict_EA or predict_DA (default).
<code>noFolds</code>	Number of folds for k-fold cross validation.
<code>reps</code>	Number of repetitions for cross validation.
<code>cores</code>	Number of cores for parallelization.

Value

A matrix of the estimated loss and results of significance testing.

References

Sharma, P. N., Liengaard, B. D., Hair, J. F., Sarstedt, M., & Ringle, C. M. (2022). Predictive model assessment and selection in composite-based modeling using PLS-SEM: extensions and guidelines for using CVPAT. *European journal of marketing*, 57(6), 1662-1677.

Liengaard, B. D., Sharma, P. N., Hult, G. T. M., Jensen, M. B., Sarstedt, M., Hair, J. F., & Ringle, C. M. (2021). Prediction: coveted, yet forsaken? Introducing a cross-validated predictive ability test in partial least squares path modeling. *Decision Sciences*, 52(2), 362-392.

Examples

```

# Load libraries
library(seminr)
library(seminrExtras)

# Create measurement model ----
corp_rep_mm_ext <- constructs(
  composite("QUAL", multi_items("qual_", 1:8), weights = mode_B),

```

```

    composite("PERF", multi_items("perf_", 1:5), weights = mode_B),
    composite("CSOR", multi_items("csor_", 1:5), weights = mode_B),
    composite("ATTR", multi_items("attr_", 1:3), weights = mode_B),
    composite("COMP", multi_items("comp_", 1:3)),
    composite("LIKE", multi_items("like_", 1:3))
  )

  alt_mm <- constructs(
    composite("QUAL", multi_items("qual_", 1:8), weights = mode_B),
    composite("PERF", multi_items("perf_", 1:5), weights = mode_B),
    composite("CSOR", multi_items("csor_", 1:5), weights = mode_B),
    composite("COMP", multi_items("comp_", 1:3)),
    composite("LIKE", multi_items("like_", 1:3))
  )

  # Create structural model ----

  corp_rep_sm_ext <- relationships(
    paths(from = c("QUAL", "PERF", "CSOR", "ATTR"), to = c("COMP", "LIKE"))
  )
  alt_sm <- relationships(
    paths(from = c("QUAL", "PERF", "CSOR"), to = c("COMP", "LIKE"))
  )

  # Estimate the model ----
  established_model <- estimate_pls(
    data = corp_rep_data,
    measurement_model = corp_rep_mm_ext,
    structural_model = corp_rep_sm_ext,
    missing = mean_replacement,
    missing_value = "-99")

  alternative_model <- estimate_pls(
    data = corp_rep_data,
    measurement_model = alt_mm,
    structural_model = alt_sm,
    missing = mean_replacement,
    missing_value = "-99")

  # Function to compare the Loss of two models
  assess_cvpat_compare(established_model,
    alternative_model ,
    testtype = "two.sided",
    nboot = 20,
    seed = 123,
    technique = predict_DA,
    noFolds = 5,
    reps = 1,
    cores = 1)

```

congruence_test	<i>SEMinR function to bootstrap calculate the congruence coefficient</i>
-----------------	--

Description

‘congruence_test’ conducts a single bootstrapped congruence test.

Usage

```
congruence_test(
  seminr_model,
  nboot = 2000,
  seed = 123,
  alpha = 0.05,
  threshold = 1
)
```

Arguments

seminr_model	The SEMinR model for CVPAT analysis
nboot	The number of bootstrap subsamples to execute (defaults to 2000).
seed	The seed for reproducibility (defaults to 123).
alpha	The required level of alpha for statistical testing (defaults to 0.05).
threshold	The threshold with which to compare significance testing $H_0: rc < 1$ (defaults to 1).

Value

A matrix of the estimated congruence coefficient and results of significance testing.

References

Franke, G. R., Sarstedt, M., & Danks, N. P. (2021). Assessing measure congruence in nomological networks. *Journal of Business Research*, 130, 318-334.

Examples

```
# Load libraries
library(seminr)
library(seminrExtras)

# Create measurement model ----
corp_rep_mm <- constructs(
  composite("QUAL", multi_items("qual_", 1:8), weights = mode_B),
  composite("PERF", multi_items("perf_", 1:5), weights = mode_B),
  composite("CSOR", multi_items("csor_", 1:5), weights = mode_B),
  composite("ATTR", multi_items("attr_", 1:3), weights = mode_B),
```

```
    composite("COMP", multi_items("comp_", 1:3)),
    composite("LIKE", multi_items("like_", 1:3))
  )

# Create structural model ----
corp_rep_sm <- relationships(
  paths(from = c("QUAL", "PERF", "CSOR", "ATTR"), to = c("COMP", "LIKE"))
)

# Estimate the model ----
corp_rep_pls_model <- estimate_pls(
  data = corp_rep_data,
  measurement_model = corp_rep_mm,
  structural_model = corp_rep_sm,
  missing = mean_replacement,
  missing_value = "-99")

# Assess the base model ----
congruence_test(seminr_model = corp_rep_pls_model,
  nboot = 20,
  seed = 123,
  alpha = 0.05,
  threshold = 1)
```

Index

`assess_cvpat`, [2](#)

`assess_cvpat_compare`, [3](#)

`congruence_test`, [6](#)