

Package ‘semtree’

April 29, 2021

Type Package

Title Recursive Partitioning for Structural Equation Models

Author Andreas M. Brandmaier [aut, cre],
John J. Prindle [aut],
Manuel Arnold [aut]

Maintainer Andreas M. Brandmaier <andy@brandmaier.de>

Depends R (>= 2.10), OpenMx (>= 2.6.9),

Imports bitops, sets, digest, rpart, rpart.plot (>= 3.0.6), parallel,
plotrix, cluster, stringr, lavaan, expm, ggplot2, viridis,
tidyr, methods, strucchange, sandwich, zoo, crayon, clisymbols,
testthat

Suggests knitr, rmarkdown, MASS, psychTools

Description SEM Trees and SEM Forests -- an extension of model-based decision trees and forests to Structural Equation Models (SEM). SEM trees hierarchically split empirical data into homogeneous groups sharing similar data patterns with respect to a SEM by recursively selecting optimal predictors of these differences. SEM forests are an extension of SEM trees. They are ensembles of SEM trees each built on a random sample of the original data. By aggregating over a forest, we obtain measures of variable importance that are more robust than measures from single trees.

License GPL-3

Encoding UTF-8

LazyLoad yes

Version 0.9.15

Date 2021-04-26

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-28 23:20:08 UTC

R topics documented:

biodiversity	2
coef.semtree	3
computePval_maxLR	3
diversityMatrix	4
evaluate	5
evaluateDataLikelihood	5
evaluateTree	6
findOtherSplits	7
fitSubmodels	8
getDepth	8
getHeight	9
getLeafs	10
getNodeById	10
getNumNodes	11
getTerminalNodes	11
isLeaf	12
kl	12
lgcm	13
merge.semforest	13
modelEstimates	14
outliers	14
parameters	15
partialDependence	16
proximity	16
prune	17
se	18
semforest	19
semforest.control	20
semtree	21
semtree.constraints	23
semtree.control	24
subforest	27
subtree	27
toTable	28
varimp	29

Index	31
--------------	-----------

biodiversity

Quantify bio diversity of a SEM Forest

Description

A function to calculate biodiversity of a `semforest` object.

Usage

```
biodiversity(x, aggregate.fun = median)
```

Arguments

`x` A [semforest](#) object

`aggregate.fun` Takes a function to apply to the vector of pairwise diversities. By default, this is the median.

Author(s)

Andreas M. Brandmaier

coef.semtree	<i>Return the parameter estimates of a given leaf of a SEM tree</i>
--------------	---

Description

Return the parameter estimates of a given leaf of a SEM tree

Usage

```
## S3 method for class 'semtree'
coef(object, ...)
```

Arguments

`object` semtree. A SEM tree node.

`...` Extra arguments. Currently unused.

`@exportS3Method coef semtree`

computePval_maxLR	<i>Wrapper function for computing the maxLR corrected p value from strucchange</i>
-------------------	--

Description

Wrapper function for computing the maxLR corrected p value from strucchange

Usage

```
computePval_maxLR(maxLR, q, covariate, from, to, nrep)
```

Arguments

maxLR	maximum of the LR test statistics
q	number of free SEM parameters / degrees of freedom
covariate	covariate under evaluation. This is important to get the level of measurement from the covariate and the bin size for ordinal and categorical covariates.
from	numeric from interval (0, 1) specifying start of trimmed sample period. With the default from = 0.15 the first and last 15 percent of observations are trimmed. This is only needed for continuous covariates.
to	numeric from interval (0, 1) specifying end of trimmed sample period. By default, to is 1.
nrep	numeric. Number of replications used for simulating from the asymptotic distribution (passed to efpFunctional). Only needed for ordinal covariates.

Value

Numeric. p value for maximally selected LR statistic

Author(s)

Manuel Arnold

diversityMatrix *Diversity Matrix*

Description

Computes a diversity matrix using a distance function between trees

Usage

```
diversityMatrix(forest, divergence = klsym, showProgressBar = TRUE)
```

Arguments

forest	A SEM forest
divergence	A divergence function such as hellinger or klsym
showProgressBar	Boolean. Show a progress bar.

evaluate	<i>Average Deviance of a Dataset given a Forest</i>
----------	---

Description

Evaluates the average deviance (-2LL) of a dataset given a forest.

Usage

```
evaluate(x, data = NULL, ...)
```

Arguments

x	A fitted semforest object
data	A data.frame
...	No extra parameters yet.

Value

Average deviance

Author(s)

Andreas M. Brandmaier

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[evaluateDataLikelihood](#), [semtree](#), [semforest](#)

evaluateDataLikelihood	<i>Compute the Negative Two-Loglikelihood of some data given a model (either OpenMx or lavaan)</i>
------------------------	--

Description

This helper function is used in the [semforest varimp](#) and [proximity](#) aggregate functions.

Usage

```
evaluateDataLikelihood(model, data, data_type = "raw")
```

Arguments

model	A OpenMx model as used in semtree and semforest .
data	Data set to apply to a fitted model.
data_type	Type of data ("raw", "cov", "cor")

Value

Returns a -2LL model fit for the model

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#), [semforest](#)

evaluateTree

Evaluate Tree -2LL

Description

A helper function to evaluate the negative two log-likelihood (-2LL) of leaf (terminal) nodes for a dataset. When given a [semtree](#) and a unique dataset, the model estimates -2LL for the tree parameters and data subsets that fit the tree branching criteria.

Usage

```
evaluateTree(tree, test_set, data_type = "raw", leaf_ids = NULL)
```

Arguments

tree	A fitted semtree object
test_set	Dataset to fit to a fitted semtree object
data_type	type of data ("raw", "cov", "cor")
leaf_ids	Identifies which nodes are leaf nodes. Default is NULL, which checks model for leaf nodes and fills this information in automatically.

Value

A list with two elements:

deviance	Combined -2LL for leaf node models of the tree.
num_models	Number of leaf nodes used for the deviance calculations.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[evaluateDataLikelihood](#), [semtree](#), [semforest](#)

findOtherSplits	<i>Find Other Node Split Values</i>
-----------------	-------------------------------------

Description

Search tool to search nodes for alternative splitting values found during the [semtree](#) process. Given a particular node, competing split values are listed assuming they also meet the criteria for a significant splitting value as set by [semtree.control](#).

Usage

```
findOtherSplits(node, tree)
```

Arguments

node	A node from a semtree object.
tree	A semtree object which the node is part of.

Value

A `data.frame()` with rows corresponding to the variable names and split values for alternative splits found in the node of interest. ...

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

fitSubmodels	<i>Fit multigroup model for evaluating a candidate split</i>
--------------	--

Description

Fit multigroup model for evaluating a candidate split

Usage

```
fitSubmodels(
  model,
  subset1,
  subset2,
  control,
  invariance = NULL,
  return.models = FALSE
)
```

Arguments

model	A model specification that is used as template for each of the two groups
subset1	Dataset for the first group model
subset2	Dataset for the second group model
control	a <code>semtree.control</code> object
invariance	fit models with invariant parameters if given. NULL otherwise (default).
return.models	boolean. Return the fitted models returns NA if fit fails

getDepth	<i>Get the depth (or, height) a tree.</i>
----------	---

Description

Returns the length of the longest path from a root node to a leaf node.

Usage

```
getDepth(tree)
```


Arguments

tree A *semtree* object

Author(s)

Andreas M. Brandmaier

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

getHeight *Determine Height of a Tree*

Description

Returns height of a SEM Tree, which equals to the length of the longest path from root to a terminal node.

Usage

getHeight(tree)

Arguments

tree A SEM tree.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

getLeafs *Get a list of all leafs in a tree*

Description

Get a list of all leafs in a tree by recursively searching the tree starting at the given node (if not data object is given. If data is given, the function returns the leafs that are predicted for each row of the given data.

Usage

```
getLeafs(tree, data = NULL)
```

Arguments

tree	A semtree object
data	A <code>data.frame</code>

Author(s)

Andreas M. Brandmaier

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

getNodeById *Get Node By Id*

Description

Return a node matching a given node ID

Usage

```
getNodeById(tree, id)
```

Arguments

tree	A SEM Tree object.
id	Numeric. A Node id.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

getNumNodes	<i>Tree Size</i>
-------------	------------------

Description

Counts the number of nodes in a tree.

Usage

```
getNumNodes(tree)
```

Arguments

tree A SEM tree object.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

getTerminalNodes	<i>Returns all leafs of a tree</i>
------------------	------------------------------------

Description

Returns all leafs (=terminal nodes) of a tree.

Usage

```
getTerminalNodes(tree)
```

Arguments

tree A semtree object.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

isLeaf	<i>Test whether a semtree object is a leaf.</i>
--------	---

Description

Tests whether a semtree object is a leaf. Returns TRUE or FALSE.

Usage

```
isLeaf(tree)
```

Arguments

tree	A semtree object
------	----------------------------------

Author(s)

Andreas M. Brandmaier

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

k1	<i>Distances</i>
----	------------------

Description

Divergence measures for multivariate normal distributions as used in the diversityMatrix function.

Usage

```
k1(mu1, cov1, mu2, cov2)
```

Arguments

mu1	Mean vector
cov1	Covariance matrix
mu2	Mean vector
cov2	Covariance matrix

lgcm	<i>Simulated Linear Latent Growth Curve Data</i>
------	--

Description

This data set provides simple data to fit with a LGCM.

Format

lgcm is a matrix containing 400 rows and 8 columns of simulated data. Longitudinal observations are o1-o5. Covariates are agegroup, training, and noise.

Author(s)

Andreas M. Brandmaier <brandmaier@mpib-berlin.mpg.de>

merge.semforest	<i>Merge two SEM forests</i>
-----------------	------------------------------

Description

This overrides generic base::merge() to merge two forests into one.

Usage

```
## S3 method for class 'semforest'  
merge(x, y, ...)
```

Arguments

x	A SEM Forest
y	A second SEM Forest
...	Extra arguments. Currently unused.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[sentree](#)

modelEstimates	<i>Returns all estimates of a tree</i>
----------------	--

Description

Return model estimates of the tree.

Usage

```
modelEstimates(tree, ...)
```

Arguments

tree	A semtree object.
...	Optional arguments.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

outliers	<i>Find outliers based on case proximity</i>
----------	--

Description

Compute outlier score based on proximity matrix.

Usage

```
outliers(prox)
```

Arguments

prox	A proximity matrix.
------	---------------------

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[proximity](#)

parameters	<i>SEMtrees Parameter Estimates Table</i>
------------	---

Description

Returns a table of parameters with columns corresponding to freely estimated parameters and rows corresponding to nodes in the tree.

Usage

```
parameters(tree, leafs.only = TRUE)
```

Arguments

tree	A SEMtree object obtained from semtree
leafs.only	Default = TRUE. Only the terminal nodes (leafs) are printed. If set to FALSE, all node parameters are written to the <code>data.frame</code> .

Details

The row names of the resulting data frame correspond to internal node ids and the column names correspond to parameters in the SEM. Standard errors of the estimates can be obtained from [parameters](#).

Value

Returns a `data.frame` with rows for parameters and columns for terminal nodes.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#), [semtree.control](#), [se](#)

partialDependence	<i>Partial Dependence Plot</i>
-------------------	--------------------------------

Description

Partial dependence plot for the effect of an independent variable in the dataset on the dependent outcome parameter selected. Returns a partialDependence object that can be plotted via generic plot command.

Usage

```
partialDependence(
  forest,
  reference.var,
  reference.param,
  support = NULL,
  cluster = NULL
)
```

Arguments

forest	A SEM forest
reference.var	Label of the (independent) reference variable for which partial dependence is plotted
reference.param	Label of the (dependent) model parameter for which partial dependence is plotted
support	Number of grid points for interpolating the reference.var
cluster	A reference to a cluster from parallel package for parallel execution. Defaults to NULL for sequential computation.

Author(s)

Andreas M. Brandmaier

proximity	<i>SEM Forest Case Proximity</i>
-----------	----------------------------------

Description

A `semforest` process to represent proportion of trees where each case are in the same leaf nodes. The values are bounded (0,1), where higher values are closer in proximity.

Usage

```
proximity(forest, dataset = NULL, type = 0, aggregate = T, cluster = NULL, ...)
```

Arguments

forest	A semforest object.
dataset	A dataset to compute proximity values for.
type	Missingness accounted for. (0 = no, 1 = yes)
aggregate	Boolean marker to compute aggregate proximity scores.
cluster	An object of class "cluster" representing a parallel socket cluster. See package makeCluster .
...	Optional arguments.

Value

A matrix with dimensions NxN is returned. The values of each cell are bounded (0,1) and represent proportion of trees where each case are in the same leaf nodes.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semforest](#), [semtree](#)

 prune

Prune a SEM Tree or SEM Forest

Description

Returns a new tree with a maximum depth selected by the user. can be used in conjunction with plot commands to view various pruning levels.

Usage

```
prune(object, ...)
```

Arguments

object A [semtree](#) or semforest object.
 ... Optional parameters, such as `max.depth` the maximum depth of each tree, or
 also `num.trees` when pruning a forest.

Details

The returned tree is only modified by the number of levels for the tree. This function does not reevaluate the data, but provides alternatives to reduce tree complexity. If the user would like to alter the tree by increasing depth, then `max.depth` option must be adjusted in the [semtree.control](#) object (provided further splits are able to be computed).

Value

Returns a [semtree](#) object.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#), [semtree.control](#)

 se

SEMtrees Parameter Estimates Standard Error Table

Description

Returns a table of standard errors with columns corresponding to freely estimated standard errors and rows corresponding to nodes in the tree.

Usage

```
se(tree, leafs.only = TRUE)
```

Arguments

tree A SEMtree object obtained from [semtree](#)
 leafs.only Default = TRUE. Only the terminal nodes (leafs) are printed. If set to FALSE,
 all node standard errors are written to the `data.frame`.

Details

The row names of the resulting data frame correspond to internal node ids and the column names correspond to standard errors in the SEM. Parameter estimates can be obtained from [parameters](#).

Value

Returns a data.frame with rows for parameters and columns for terminal nodes.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#), [semtree.control](#), [parameters](#)

 semforest

Create a SEM Forest

Description

Grows a SEM Forest from a template model and a dataset. This may take some time.

Usage

```
semforest(
  model,
  data,
  control = NULL,
  predictors = NULL,
  constraints = NULL,
  cluster = NULL,
  ...
)
```

Arguments

model	A template SEM. Same as in <code>semtree</code> .
data	A dataframe to create a forest from. Same as in <code>semtree</code> .
control	A <code>semforest</code> control object to set forest parameters.
predictors	An optional list of covariates. See <code>semtree</code> code example.

constraints	An optional list of covariates. See <code>semtree</code> code example.
cluster	An object of class "cluster" representing a parallel socket cluster. See package makeCluster .
...	Optional parameters.

Value

A `semforest` object.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Prindle, J. J., McArdle, J. J., & Lindenberger, U. (2016). Theory-guided exploration with structural equation model forests. *Psychological Methods*, 21(4), 566–582.

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71–86.

See Also

[semtree](#)

<code>semforest.control</code>	<i>SEM Forest Control Object</i>
--------------------------------	----------------------------------

Description

A SEM Forest control object to tune parameters of the forest learning algorithm.

Usage

```
semforest.control(
  num.trees = 5,
  sampling = "subsample",
  control = NA,
  mtry = 2
)
```

Arguments

<code>num.trees</code>	Number of trees.
<code>sampling</code>	Sampling procedure. Can be <code>subsample</code> or <code>bootstrap</code> .
<code>control</code>	A SEM Tree control object. Will be generated by default.
<code>mtry</code>	Number of subsampled covariates at each node.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

 semtree

SEM Tree: Recursive Partitioning for Structural Equation Models

Description

Structural equation model (SEM) trees are a combination of SEM and decision trees (also known as classification and regression trees or recursive partitioning). SEM trees hierarchically split empirical data into homogeneous groups sharing similar data patterns with respect to a SEM by recursively selecting optimal predictors of these differences from a potentially large set of predictors.

Usage

```
semtree(
  model,
  data = NULL,
  control = NULL,
  constraints = NULL,
  predictors = NULL,
  ...
)
```

Arguments

model	A template model specification from OpenMx using the <code>mxModel</code> function (or a lavaan model using the <code>lavaan</code> function with option <code>fit=FALSE</code>). Model must be syntactically correct within the framework chosen, and converge to a solution.
data	<code>Data.frame</code> used in the model creation using <code>mxModel</code> or <code>lavaan</code> are input here. Order of modeled variables and predictors is not important when providing a dataset to <code>semtree</code> .
control	<code>semtree</code> model specifications from <code>semtree.control</code> are input here. Any changes from the default setting can be specified here.
constraints	A <code>semtree.constraints</code> object setting model parameters as constrained from the beginning of the <code>semtree</code> computation. This includes options to globally or locally set equality constraints and to specify focus parameters (i.e., parameter subsets that exclusively go into the function evaluating splits). Also, options for measurement invariance testing in trees are included.

predictors	A vector of variable names matching variable names in dataset. If NULL (default) all variables that are in dataset and not part of the model are potential predictors. Optional function input to select a subset of the unmodeled variables to use as predictors in the semtree function.
...	Optional arguments passed to the tree growing function.

Details

Calling `semtree` with an [OpenMx](#) or [lavaan](#) model creates a tree that recursively partitions a dataset such that the partitions maximally differ with respect to the model-predicted distributions. Each resulting subgroup (represented as a leaf in the tree) is represented by a SEM with a distinct set of parameter estimates.

Predictors (yet unmodeled variables) can take on any form for the splitting algorithm to function (categorical, ordered categories, continuous). Care must be taken in choosing how many predictors to include in analyses because as the number of categories grows for unordered categorical variables, the number of multigroup comparisons increases exponentially for unordered categories.

Currently available evaluation methods for assessing partitions:

1. "naive" selection method compares all possible split values to one another over all predictors included in the dataset.
2. "fair" selection uses a two step procedure for analyzing split values on predictors at each node of the tree. The first phase uses half of the sample to examine the model improvement for each split value on each predictor, and retains the the value that presents the largest improvement for each predictor. The second phase then evaluates these best split points for each predictor on the second half of the sample. The best improvement for the c splits tested on c predictors is selected for the node and the dataset is split from this node for further testing.
3. "score" uses score-based test statistics. These statistics are much faster than the classic SEM tree approach while having favorable statistical properties.

All other parameters controlling the tree growing process are available through a separate [semtree.control](#) object.

Value

A `semtree` object is created which can be examined with `summary`, `plot`, and `print`.

Author(s)

Andreas M. Brandmaier, John J. Prindle, Manuel Arnold

References

- Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.
- Arnold, M., Voelke, M. C., & Brandmaier, A. M. (2021). Score-guided structural equation model trees. *Frontiers in Psychology*, 11, Article 564403. <https://doi.org/10.3389/fpsyg.2020.564403>

See Also

[semtree.control](#), [summary.semtree](#), [parameters](#), [se](#), [prune.semtree](#), [subtree](#), [OpenMx](#), [lavaan](#)

semtree.constraints *SEM Tree Constraints Object*

Description

A SEM Tree constraints object holds information regarding specifics on how the tree is grown (similar to the control object). The SEM tree control object holds all information that is independent of a specific model whereas the constraints object holds information that is specific to a certain model (e.g., specifies differential treatment of certain parameters, e.g., by holding them constant across the forest).

Usage

```
semtree.constraints(  
  local.invariance = NULL,  
  global.invariance = NULL,  
  focus.parameters = NULL  
)
```

Arguments

local.invariance

Vector of parameter names that are locally equal, that is, they are assumed to be equal when assessing a local split but allowed to differ subsequently.

global.invariance

Vector of parameter names that are globally equal, that is, estimated only once and then fixed in the tree.

focus.parameters

Vector of parameter names that exclusively are evaluated for between-group differences when assessing split candidates. If NULL all parameters add to the difference.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#)

`semtree.control`*SEM Tree Control Object*

Description

A `semtree.control` object contains parameters that determine the tree growing process. These parameters include choices of different split candidate selection procedures and hyperparameters of those. Calling the constructor without parameters creates a default control object. A number of tree growing methods are included in with this package: 1. "naive" splitting takes the best split value of all possible splits on each covariate. 2. "fair" selection is so called because it tests all splits on half of the data, then tests the best split value for each covariate on the other half of the data. The equal footing of each covariate in this two phase test removes bias from testing variables with many possible splits compared to those with few. 3. "fair3" does the phases described above, with an additional step of retesting all of the split values on the best covariate found in the second phase. Variations in the sample from subsetting are removed and bias in split selection further reduced. 4. "crossvalidation" partitions the data for maximizing splits on each variable, then comparing maximum splits across each variable on the rest of the data.

Usage

```
semtree.control(  
  method = "naive",  
  min.N = 20,  
  max.depth = NA,  
  alpha = 0.05,  
  alpha.invariance = NA,  
  folds = 5,  
  exclude.heywood = TRUE,  
  progress.bar = TRUE,  
  verbose = FALSE,  
  bonferroni = FALSE,  
  use.all = FALSE,  
  seed = NA,  
  custom.stopping.rule = NA,  
  mtry = NA,  
  report.level = 0,  
  exclude.code = NA,  
  score.tests = list(nominal = "LMuo", ordinal = "maxLMo", metric = "maxLM"),  
  information.matrix = "info",  
  scaled_scores = TRUE,  
  linear = TRUE,  
  min.bucket = 10,  
  naive.bonferroni.type = 0,  
  missing = "ignore",  
  use.maxlm = FALSE,  
  strucchange.from = 0.15,  
  strucchange.to = NULL,
```



```

    strucchange.nrep = 50000
  )

```

Arguments

method	Default: "naive". One out of <code>c("fair", "fair3", "naive", "cv")</code> for either an unbiased two-step selection algorithm, three-step fair algorithm, a naive take-the-best, or a cross-validation scheme.
min.N	Default: 10. Minimum sample size per a node, used to determine whether to continue splitting a tree or establish a terminal node.
max.depth	Default: NA. Maximum levels per a branch. Parameter for limiting tree growth.
alpha	Default: 0.05. Significance level for splitting at a given node.
alpha.invariance	Default: NA. Significance level for invariance tests. If NA, the value of alpha is used.
folds	Default: 5. Defines the number of folds for the "cv" method.
exclude.heywood	Default: TRUE. Reports whether there is an identification problem in the covariance structure of an SEM tested.
progress.bar	Default: NA. Option to disable the progress bar for tree growth.
verbose	Default: FALSE. Option to turn on or off <i>all</i> model messages during tree growth.
bonferroni	Default: FALSE. Correct for multiple tests with Bonferroni type correction.
use.all	Treatment of missing variables. By default, missing values stay in a decision node. If TRUE, cases are distributed according to a maximum likelihood principle to the child nodes.
seed	Default: NA. Set a random number seed for repeating random fold generation in tree analysis.
custom.stopping.rule	Default: NA. Otherwise, this can be a boolean function with a custom stopping rule for tree growing.
mtry	Default: NA. Number of sample columns to use in SEMforest analysis.
report.level	Default: 0. Values up to 99 can be used to increase the number of onscreen reports for semtree analysis.
exclude.code	Default: NA. NPSOL error code for exclusion from model fit evaluations when finding best split. Default: Models with errors during fitting are retained.
score.tests	A list of score-based test statistics from the strucchange package to be used for different variable types.
information.matrix	A function to extract the covariance matrix for the coefficients of the fitted model.
scaled_scores	If TRUE (default), a scaled cumulative score process is used for identifying a cutpoint.

linear	If TRUE (default), the structural equation model is assumed to be linear without any nonlinear parameter constraints. The runtime is much smaller for linear MxRAM-type models than for models with nonlinear constraints on the parameters.
min.bucket	Minimum bucket size to continue splitting
naive.bonferroni.type	Default: 0. When set to zero, bonferroni correction for the naive test counts the number of dichotomous tests. When set to one, bonferroni correction counts the number of variables tested.
missing	Missing value treatment. Default is ignore
use.maxlm	Use MaxLm statistic
strucchange.from	Strucchange argument. See their package documentation.
strucchange.to	Strucchange argument. See their package documentation.
strucchange.nrep	Strucchange argument. See their package documentation.

Value

A control object containing a list of the above parameters.

Author(s)

Andreas M. Brandmaier, John J. Prindle, Manuel Arnold

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#)

Examples

```
# create a control object with an alpha level of 1%
my.control <- semtree.control(alpha=0.01)

# set the minimum number of cases per node to ten
my.control$min.N <- 10

# print contents of the control object
print(my.control)
```

subforest	<i>Creates subsets of trees from forests</i>
-----------	--

Description

Creates subsets of a forest. This can be used to subset a number of trees, e.g. from:(from+num), or to remove all null (type="nonnull") trees that were due to errors, or to randomly select a sub forest (type=random).

Usage

```
subforest(forest, num = NULL, type = "nonnull", from = 1)
```

Arguments

forest	A SEM Forest object.
num	Number of trees to select.
type	Either 'random' or 'nonnull' or NULL. First selects a random subset, second selects all non-null trees, third allows subsetting trees.
from	Starting index if type=NULL.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

subtree	<i>SEMtree Partitioning Tool</i>
---------	----------------------------------

Description

The subtree function returns a tree from a selected node of the [semtree](#) returned tree.

Usage

```
subtree(tree, startNode = NULL, level = 0, foundNode = FALSE)
```

Arguments

tree	A SEMtree object obtained from semtree
startNode	Node id, which will be future root node (0 to max node number of tree)
level	Ignore. Only used internally.
foundNode	Ignore. Only used internally.

Details

The row names of the resulting data frame correspond to internal node ids and the column names correspond to standard errors in the SEM. Standard errors of the estimates can be obtained from [se](#).

Value

Returns a [semtree](#) object which is a partitioned tree from the input semtree.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

See Also

[semtree](#), [semtree.control](#)

toTable	<i>Tabular Representation of a SEM Tree</i>
---------	---

Description

Converts a tree into a tabular representation. This may be useful as a textual representation for use in manuscripts.

Usage

```
toTable(tree, added.param.cols = NULL, round.param = NULL)
```

Arguments

tree	A SEM Tree object.
added.param.cols	Add extra columns with parameter estimates.
round.param	Number of digits to round parameter estimates

Author(s)

Andreas M. Brandmaier

References

Brandmaier, A. M., Ram, N., Wagner, G. G., & Gerstorf, D. (in press). Terminal decline in well-being: The role of multi-indicator constellations of physical health and psychosocial correlates. *Developmental Psychology*.

varimp	<i>SEM Forest Variable Importance</i>
--------	---------------------------------------

Description

A function to calculate relative variable importance for selecting node splits over a [semforest](#) object.

Usage

```
varimp(
  forest,
  var.names = NULL,
  verbose = F,
  cluster = NULL,
  eval.fun = evaluateTree,
  method = "permutation",
  conditional = FALSE,
  ...
)
```

Arguments

forest	A semforest object
var.names	Covariates used in the forest creation process. NULL value will be automatically filled in by the function.
verbose	Boolean to print messages while function is running.
cluster	An object of class "cluster" representing a parallel socket cluster. See package makeCluster .
eval.fun	Default is evaluateTree function. The value of the -2LL of the leaf nodes is compared to baseline overall model.
method	Experimental. Some alternative methods to compute importance. Default is "permutation".
conditional	Conditional variable importance if TRUE, otherwise marginal variable importance.
...	Optional arguments.

Author(s)

Andreas M. Brandmaier, John J. Prindle

References

Brandmaier, A.M., Oertzen, T. v., McArdle, J.J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, 18(1), 71-86.

Index

- * **biodiversity**
 - biodiversity, 2
- * **datasets**
 - lgcm, 13
- * **models**
 - semforest, 19
 - semtree, 21
- * **multivariate**
 - semforest, 19
 - semtree, 21
- * **semforest**
 - biodiversity, 2
- * **tree**
 - semforest, 19
 - semtree, 21

biodiversity, 2

coef.semtree, 3

computePval_maxLR, 3

diversityMatrix, 4

evaluate, 5

evaluateDataLikelihood, 5, 5, 7

evaluateTree, 6, 29

findOtherSplits, 7

fitSubmodels, 8

getDepth, 8

getHeight, 9

getLeafs, 10

getNodeById, 10

getNumNodes, 11

getTerminalNodes, 11

hellinger (kl), 12

isLeaf, 12

kl, 12

klsym (kl), 12

lavaan, 21–23

lgcm, 13

makeCluster, 17, 20, 29

merge.semforest, 13

modelEstimates, 14

mxModel, 21

nodeFunSemtree (semtree), 21

OpenMx, 6, 21–23

outliers, 14

parameters, 15, 15, 19, 23

partialDependence, 16

partialDependencePlot

- (partialDependence), 16

plot.partialDependence

- (partialDependence), 16

plot.proximity (proximity), 16

plot.semforest (semforest), 19

plot.semforest.proximity (proximity), 16

plot.semforest.varimp (varimp), 29

plot.semtree (semtree), 21

print.semforest (semforest), 19

print.semforest.control

- (semforest.control), 20

print.semforest.varimp (varimp), 29

print.semtree (semtree), 21

print.semtree.control

- (semtree.control), 24

proximity, 5, 15, 16

prune, 17

prune.semtree, 23

se, 15, 18, 23, 28

semforest, 2, 3, 5–7, 16, 17, 19, 29

semforest.control, 20

semtree, [5–7](#), [9](#), [10](#), [12](#), [13](#), [15](#), [17–21](#), [21](#), [23](#),
[26–28](#)
semtree.constraints, [21](#), [23](#)
semtree.control, [7](#), [15](#), [18](#), [19](#), [21–23](#), [24](#), [28](#)
subforest, [27](#)
subtree, [23](#), [27](#)
summary.semtree, [23](#)
summary.semtree (semtree), [21](#)

toLatex.semtree (semtree), [21](#)
toTable, [28](#)

varimp, [5](#), [29](#)
varimpConvergencePlot (varimp), [29](#)