

Package ‘semtree’

February 13, 2018

Type Package

Title Recursive Partitioning for Structural Equation Models

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

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

Depends OpenMx (>= 2.6.9)

Imports bitops, sets, digest, rpart, rpart.plot, parallel, plotrix,
cluster

Suggests lavaan

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 0.9.12

Date 2018-02-13

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-13 09:48:15 UTC

R topics documented:

biodiversity 2

Distances	3
diversityMatrix	3
evaluate	4
evaluateDataLikelihood	4
evaluateTree	6
findOtherSplits	7
getHeight	7
getNodeById	8
getNumNodes	9
getTerminalNodes	9
lgcm	10
merge	10
modelEstimates	11
outliers	12
parameters	12
partialDependence	13
proximity	14
prune	15
se.semtree	16
semforest	17
semforest.control	18
semtree	18
semtree.constraints	20
semtree.control	21
subforest	23
subtree	23
toTable	24
varimp	25
Index	26

biodiversity	<i>Quantify bio diversity of a SEM Forest</i>
--------------	---

Description

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

Usage

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

Arguments

<code>x</code>	A <code>semforest</code> object
<code>aggregate.fun</code>	Takes a function to apply to the vector of pairwise diversities. By default, this is the median.

Author(s)

Andreas M. Brandmaier

*Distances**Distances*

Description

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

Usage

```
klsym(mu1, cov1, mu2, cov2)
hellinger(mu1, cov1, mu2, cov2)
```

Arguments

<code>mu1</code>	Mean vector
<code>mu2</code>	Mean vector
<code>cov1</code>	Covariance matrix
<code>cov2</code>	Covariance matrix

`diversityMatrix`*Diversity Matrix*

Description

Computes a diversity matrix using a distance function between trees

Usage

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

Arguments

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

evaluate

Average Deviance of a Dataset given a Forest

Description

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

Usage

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

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

Evaluate Data Likelihood

Description

Evaluate Fitted Model Likelihood for Dataset. 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](#)

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (model, data, data_type = "raw")
{
  model <- omxSetParameters(model, labels = names(omxGetParameters(model)),
    free = F)
  if (data_type == "raw") {
    data <- full_mxdata <- mxData(observed = data, type = "raw")
  }
  else if (data_type == "cov") {
    data <- full_mxdata <- mxData(observed = cov(data), type = "cov",
      numObs = dim(data)[1])
  }
  else if (data_type == "cor") {
    data <- full_mxdata <- mxData(observed = cor(data), type = "cor",
      numObs = dim(data)[1])
  }
  else {
    warning("data type is not supported!")
    return(NULL)
  }
  model <- setData(model, data)
  run <- OpenMx::mxRun(model, silent = T, useOptimizer = F,
    suppressWarnings = T)
```

```

    result <- OpenMx::mxEval(objective, run)
    return(result)
  }

```

 evaluateTree

Evaluate Tree -2LL

Description

A helper function to evaluate the -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.

getHeight	<i>Determine Height of a Tree</i>
-----------	-----------------------------------

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.

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.

lgcm

Simulated Linear Latent Growth Curve Data

Description

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

Usage

```
data(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

Merge two SEM forests

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

[semtree](#)

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

Description

Return model estimates of the tree.

Usage

```
modelEstimates(tree, level = 0, ...)
```

Arguments

tree	A semtree object.
level	
...	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 *Partial Dependence Plot*

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 *SEM Forest Case Proximity*

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.semtree`*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

<code>tree</code>	A SEMtree object obtained from semtree
<code>leafs.only</code>	Default = TRUE. Only the terminal nodes (leafs) are printed. If set to FALSE, all node standard errors 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 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	<i>Create a SEM Forest</i>
-----------	----------------------------

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](#)

semforest.control *SEM Forest Control Object*

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

num.trees	Number of trees.
sampling	Sampling procedure. Can be subsample or bootstrap.
control	A SEM Tree control object. Will be generated by default.
mtry	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.

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. "fair3" is a a variant of the previous method with an additional phase. The first two phases are done as described above. Then in the third phase, the complete sample is recombined and used to evaluate the best split point on
4. "crossvalidation" recursion method uses the "folds" option in `semtree.control` to partition the data into a specified number of subsamples (default = 5). to change this setting use `semtree.control(folds=n)` to change the number of folds to value n.

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	Data.frame 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 <code>semtree</code> function.
...	Optional arguments passed to the tree growing function.

Details

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

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.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=.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 )
```

Arguments

<code>method</code>	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.
<code>min.N</code>	Default: 10. Minimum sample size per a node, used to determine whether to continue splitting a tree or establish a terminal node.
<code>max.depth</code>	Default: NA. Maximum levels per a branch. Parameter for limiting tree growth.
<code>alpha</code>	Default: 0.05. Significance level for splitting at a given node.
<code>alpha.invariance</code>	Default: NA. Significance level for invariance tests. If NA, the value of <code>alpha</code> is used.

fold	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.
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.
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.
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.
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.

Value

A control object containing a list of the above parameters.

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](#)

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

Tabular Representation of a SEM Tree

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", ...)
```

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.
method	Experimental. Some alternative methods to compute importance. Default is "permutation".
eval.fun	Default is evaluateTree function. The value of the -2LL of the leaf nodes is compared to baseline overall model.
cluster	An object of class "cluster" representing a parallel socket cluster. See package makeCluster .
...	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

*Topic **\textasciitildekwd1**

- evaluateDataLikelihood, 4
- evaluateTree, 6
- getNodeById, 8
- getNumNodes, 9
- getTerminalNodes, 9
- outliers, 12
- proximity, 14
- subforest, 23
- varimp, 25

*Topic **\textasciitildekwd2**

- evaluateDataLikelihood, 4
- evaluateTree, 6
- getNodeById, 8
- getNumNodes, 9
- getTerminalNodes, 9
- outliers, 12
- proximity, 14
- subforest, 23
- varimp, 25

*Topic **biodiversity**

- biodiversity, 2

*Topic **datasets**

- lgcm, 10

*Topic **models**

- semforest, 17
- semtree, 18

*Topic **multivariate**

- semforest, 17
- semtree, 18

*Topic **semforest**

- biodiversity, 2

*Topic **tree**

- semforest, 17
- semtree, 18

biodiversity, 2

Distances, 3

diversityMatrix, 3

evaluate, 4

evaluateDataLikelihood, 4, 4, 6

evaluateTree, 6, 25

findOtherSplits, 7

getHeight, 7

getNodeById, 8

getNumNodes, 9

getTerminalNodes, 9

hellinger (Distances), 3

klsym (Distances), 3

lavaan, 18–20

lgcm, 10

makeCluster, 14, 17, 25

merge, 10

modelEstimates, 11

mxModel, 19

nodeFunSemtree (semtree), 18

OpenMx, 5, 18–20

outliers, 12

parameters, 12, 13, 16, 20

partialDependence, 13

partialDependencePlot
(partialDependence), 13

plot.partialDependence
(partialDependence), 13

plot.proximity (proximity), 14

plot.semforest (semforest), 17

plot.semforest.proximity (proximity), 14

plot.semforest.varimp (varimp), 25

plot.semtree (semtree), 18

print.semforest (semforest), 17

print.semforest.control
(semforest.control), 18

`print.semforest.varimp` (`varimp`), 25
`print.semtree` (`semtree`), 18
`print.semtree.control`
 (`semtree.control`), 21
`proximity`, 4, 12, 14
`prune`, 15
`prune.semtree`, 20

`se`, 13, 20, 24
`se` (`se.semtree`), 16
`se.semtree`, 16
`semforest`, 2, 4–6, 14, 17, 25
`semforest.control`, 18
`semtree`, 4–7, 11–17, 18, 19, 21–24
`semtree.constraints`, 19, 20
`semtree.control`, 7, 13, 15, 16, 19, 20, 21, 24
`subforest`, 23
`subtree`, 20, 23
`summary.semtree`, 20
`summary.semtree` (`semtree`), 18

`toLatex.semtree` (`semtree`), 18
`toTable`, 24

`varimp`, 4, 25
`varimpConvergencePlot` (`varimp`), 25