Package ‘C443’

April 20, 2018

Type Package
Title See a Forest for the Trees
Version 1.0.0
Imports MASS, partykit, rpart, RColorBrewer, grDevices, gridExtra,
ggplot2, cluster, parallel, igraph, reshape2, qgraph, stats,
graphics
LazyData true
Date 2018-04-18
Description Getting insight into a forest of classification trees, by calculating similarities
between the trees, and subsequently clustering them. Each cluster is represented by it's most
License GPL (>= 2)
RoxygenNote 6.0.1
NeedsCompilation no
Author Aniek Sies [aut, cre],
Iven Van Mechelen [ths]
Maintainer Aniek Sies <anieksies@kuleuven.be>
Repository CRAN
Date/Publication 2018-04-20 08:29:09 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>clusterforest</td>
<td>2</td>
</tr>
<tr>
<td>drugs</td>
<td>3</td>
</tr>
<tr>
<td>emp_eq</td>
<td>5</td>
</tr>
<tr>
<td>growforest</td>
<td>6</td>
</tr>
<tr>
<td>similarities</td>
<td>7</td>
</tr>
<tr>
<td>treesource</td>
<td>9</td>
</tr>
</tbody>
</table>

Index 10
clusterforest

Clustering the classification trees in a forest

Description

Function to cluster classification trees in a forest using Partitioning Around Medoids (PAM, Kaufman & Rousseeuw, 2009).

Usage

clusterforest(simmatrix, trees, fulldata, treedata, y, A = NULL, fromclus, toclus)

Arguments

simmatrix
Similarity matrix containing the similarities between all pairs of trees in the forest

trees
A list with all trees that should be clustered, each tree should be stored as party object

fulldata
The original full dataset

treedata
A list with data sets on which the trees in the forest were based (i.e., one data set for each tree)

y
A vector with the name of the outcome variable on which each tree in the forest was based

A
by default, in case of a treatment regime, it should denote the name of the variable that indicates the assigned treatment alternative in the data set

fromclus
The lowest number of clusters for which the clustering should be done

toclus
The highest number of clusters for which the clustering should be done

Value

medoids
the position of the medoid trees in the forest (i.e., which element of the list of trees)

mds
the medoid trees

clusters
The cluster number to which each tree is assigned

silplot
Plot of the average silhouette width for each solution

withinplot
Plot of the within cluster similarity for each solution

agreementplot
Plot of the agreement between the assignments of the forest as a whole, and those based on the medoids for each solution
Drugs

Drug consumption data set

Description

A dataset collected by Fehrman et al. (2017), freely available on the UCI Machine Learning Repository (Lichman, 2013) containing records of 1885 respondents regarding their use of 18 types of drugs, and their measurements on 12 predictors. All predictors were originally categorical and were quantified by Fehrman et al. (2017). The meaning of the values can be found on https://archive.ics.uci.edu/ml/datasets/Drug+consumption+ The original response categories for each drug were: never used the drug, used it over a decade ago, or in the last decade, year, month, week, or day. We transformed these into binary response categories, where 0 (non-user) consists of the categories never used the drug and used it over a decade ago and 1 (user) consists of all other categories.

Usage

drugs
Format

A data frame with 1185 rows and 32 variables:

ID  Respondent ID
Age  Age of respondent
Gender  Gender of respondent, where 0.48 denotes female and -0.48 denotes male
Edu  Level of education of participant
Country  Country of current residence of participant
Ethn  Ethnicity of participant
Neuro  NEO-FFI-R Neuroticism score
Extr  NEO-FFI-R Extraversion score
Open  NEO-FFI-R Openness to experience score
Agree  NEO-FFI-R Agreeableness score
Consc  NEO-FFI-R Conscientiousness score
Impul  Impulsiveness score measured by BIS-11
Sensat  Sensation seeking score measured by ImpSS
Alc  Alcohol user (1) or non-user (0)
Amphet  Amphetamine user (1) or non-user (0)
Amyl  Amyl nitrite user (1) or non-user (0)
Benzos  Benzodiazepine user (1) or non-user (0)
Caff  Caffeine user (1) or non-user (0)
Can  Cannabis user (1) or non-user (0)
Choco  Chocolate user (1) or non-user (0)
Coke  Coke user (1) or non-user (0)
Crack  Crack user (1) or non-user (0)
Ecst  Ecstasy user (1) or non-user (0)
Her  Heroin user (1) or non-user (0)
Ket  Ketamine user (1) or non-user (0)
Leg-highs  Legal Highs user (1) or non-user (0)
LSD  LSD user (1) or non-user (0)
Meth  Methadone user (1) or non-user (0)
Mush  Magical Mushroom user (1) or non-user (0)
Nico  Nicotine user (1) or non-user (0)
Semeron  Semeron user (1) or non-user (0), fictitious drug to identify over-claimers
VSA  volatile substance abuse user(1) or non-user (0)

Source

emp_eq

Check empirical equivalences between predictors

Description

Function to check for empirical equivalence relations between predictors in a data set by visualizing (partial) correlations

Usage

emp_eq(data, X)

Arguments

data Original full data set
X The names of the predictors in the data set

Value

cp Heatmap of correlations between all predictors
Graph_pcor Partial correlation network between all predictors
Graph_pcor_bon Partial correlation network between all predictors, with bonferroni correction
cormat Matrix with correlations between all variables

Examples

```r
require(MASS)
emp_eq(data = Pima.tr, X = c("npreg", "glu", "bp", "skin", "bmi", "ped", "age"))
```
growforest

Grow a forest of classification trees or tree-based treatment regimes

Description

Function to grow a forest based on (a) one data set and one outcome variable, by drawing bootstrap samples and growing a tree on each bootstrap sample, (b) one data set and multiple outcome variables, by drawing bootstrap samples and growing a tree for each outcome variable on each bootstrap sample, (c) multiple data sets and one outcome variable, by growing a tree on each data set. Trees can be either classification trees estimated using rpart or tree-based treatment regimes estimated using the method of Zhang et al (2012) with the AIPWE and rpart.

Usage

growforest(data, X, Y, ntrees, regmod = NULL, A = NULL, regime = FALSE, minsplit = 40, minbucket = 20, maxdepth = 3)

Arguments

data
The data set from which bootstrap samples should be drawn, or the data sets on which the trees should be grown

X
The names of the predictor variables in the data set that will be used as possible split variables

Y
The name of the outcome variable(s) in the data set

ntrees
The number of trees that should be grown on each data set (i.e., the number of bootstrap samples that should be drawn)

regmod
NULL by default, in case of a treatment regime, it should contain the outcome model that should be used for the augmentation

A
NULL by default, in case of a treatment regime, it should denote the name of the variable that indicates the assigned treatment alternative in the data set

regime
FALSE by default, TRUE if tree-based treatment regimes instead of classification trees are desired.

minsplit
40 by default, indicates the minimum number of observations that must exist in a node in order for a split to be attempted.

minbucket
20 by default, indicates the minimum number of observations in any terminal node.

maxdepth
3 by default, the maximum depth of any node of the final tree, with the root node counted as depth 0.

Value

partytrees
The classification trees or tree-based treatment regimes saved as party objects

Boots
The drawn bootstrap samples on which the trees/treatment regimes were based.
similarities

References


Examples

require(MASS)

# Create forest by drawing bootstrap samples and growing a tree on each bootstrap sample
forest <- growforest(data = Pima.tr, X = c("npreg", "glu", "bp", "skin", "bmi", "ped", "age"),
                     Y = "type", ntrees = 50)

# Create forest by drawing bootstrap samples and growing a tree for each outcome variable
# on each bootstrap sample
forest <- growforest(data = drugs, X = c("Age", "Gender", "Edu", "Neuro", "Extr", "Open", "Agree",
                                         "Consc", "Impul", "Sensat"),
                            "Amyl", "Ket"), ntrees = 8)

similarities

Calculating similarities between classification trees

Description

Function to calculate similarities between classification trees, based on 6 different possible similarity measures.

Usage

similarities(fulldata, treedata, y, x, trees, m, weight = NULL, a = NULL,
              tol = NULL, regime = FALSE)

Arguments

fulldata
A list with data sets on which the trees in the forest were based (i.e., one data set for each tree)

treedata
A vector with the name of the outcome variable on which each tree in the forest was based

y
The names of the predictor variables that were used as possible split variables

x
A list with all trees between which similarities should be computed, each tree should be stored as party object

trees
Similarity measure that should be used to calculate similarities, where m=1 is based on counting equal predictors or predictor-split point combinations (Equation 5 or 8 in Sies & Van Mechelen (Submitted), m=2 is the measure of Shannon & Banks (1999), based on counting the number of equal paths from rootnode to leafs (See Sies & Van Mechelen Submitted, Equation 2), m=3 is based on the

agreement in classification labels (Chipman, 1998), see Sies & Van Mechelen (submitted), Equation 14, m=4 is based on the agreement of partitions (Chipman, 1998), see Sies & Van Mechelen (Submitted), Equation 13, and m=5 is based on counting equal elementary conjunctions of trees transformed to disjunctive normal form (only for binary predictors, see Sies & Van Mechelen, Submitted, Equation 16). Finally M6 is based on comparing sets of predictor split point combinations (taking into account directions of the splits) associated with a leaf, taking into account the classification label of that leaf, see Sies & Van Mechelen (submitted).

weight

Indicating whether or not split points should be taken into account for m=1, where 0 means no (Equation 4 in Sies & Van Mechelen, submitted) and 1 means yes (Equation 8 in Sies & Van Mechelen, submitted).

A

The name of the treatment variable in case of a forest of tree-based treatment regimes, otherwise NULL by default.

tol

In case that weight = 1: A vector with for each predictor the tolerance zone within which two split points of the predictor in question are assumed equal. Default=NULL

regime

Indicating whether the trees in the forest are treatment regimes (TRUE) or decision trees (FALSE). Default=FALSE

Value

Similarity matrix based on chosen similarity measure

References


Sies, A. & Van Mechelen I. (Submitted). C443: An R-package to see a forest for the trees

Examples

```
require(MASS)
#Grow a forest of classification trees based on 10 bootstrap samples
forest <- growforest(Pima.tr, X=c("npreg", "glu", "bp", "skin", "bmi", "ped", "age"),
                    Y ="type", ntrees = 10)

# Calculate similarities between all pairs of trees in the forest
simmatrix <- similarities(fulldata = Pima.tr, treedata = forest[[2]], Y = rep("type", 10),
                          X = c("npreg", "glu", "bp", "skin", "bmi", "ped", "age"),
                          trees = forest[[1]], m = 1, weight = 0)

simmatrix2 <- similarities(fulldata = Pima.tr, treedata = forest[[2]], Y = rep("type",10),
                          X = c("npreg", "glu", "bp", "skin", "bmi", "ped", "age"),
                          trees = forest[[1]], m = 1, weight = 1, tol = c(3, 30, 10, 10, 5, 0.3, 10))
```
treesource

The number of trees of each source that belong to each cluster

Description

Function to visualize the number of trees from each source that belong to each cluster.

Usage

treesource(source, clustering)

Arguments

source A vector with the name of the source on which each tree in the forest was based
clustering A vector with the clusternumber to which each tree belongs

Value

multiplot For each outcome variable, a bar plot with the number of trees that belong to each cluster
heatmap A heatmap with for each outcome variable, the number of trees that belong to each cluster

Examples

#Grow forest based on multiple outcome variables, with 5 trees for each outcome variable

#Calculate similarities between the trees in the forest
simmatrix1 <- similarities(fulldata = drugs, treedata = forest[[2]], Y = rep(c("Amphet", "Benzos", "Coke", "Ecst"), each = 5),
trees = forest[[1]], m = 1, weight = 0)

#Cluster the trees in the forest
clusters <- clusterforest(simmatrix = simmatrix1, trees = forest[[1]], fulldata=drugs, treedata=forest[[2]], Y = rep(c("Amphet", "Benzos", "Coke", "Ecst"), each = 5), fromclus=3, toclus=3)

#Visualize the number of trees for each source that belong to each cluster
treesource(source = rep(c("Amphet", "Benzos", "Coke", "Ecst"), each = 5), clustering = clusters $ clusters[[3]])
Index

*Topic datasets
  drugs, 3
clusterforest, 2
drugs, 3
emp_eq, 5
growforest, 6
similarities, 7
treesource, 9