Package 'FFTrees'

July 11, 2017

Type Package

Title Generate, Visualise, and Evaluate Fast-and-Frugal Decision Trees

Version 1.3.3

Date 2017-7-11

Maintainer Nathaniel Phillips <Nathaniel.D.Phillips.is@gmail.com>

Description Create, visualize, and test fast-and-
frugal decision trees (FFTs). FFTs are very simple decision trees for
binary classification problems. FFTs can be preferable to more complex algorithms because they are easy to communicate, require very little information, and are
robust against overfitting.

LazyData TRUE

Depends R(>= 2.10)

Imports rpart, yarrr, circlize, parallel, graphics, randomForest,
igraph, e1071, stringr, progress

License CC0

BugReports https://github.com/ndphillips/FFTrees/issues

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Nathaniel Phillips [aut, cre],
Hansjoerg Neth [aut],
Jan Woike [aut],
Wolfgang Gaissmaer [aut]

Repository CRAN

Date/Publication 2017-07-11 10:31:35 UTC

R topics documented:

apply.tree — *Applies a fast-and-frugal tree to a dataset and generates several accuracy statistics*

Description

Applies a fast-and-frugal tree to a dataset and generates several accuracy statistics

Usage

```
apply.tree(data, formula, tree.definitions, sens.w = 0.5,
  cost.outcomes = c(0, 1, 1, 0), cost.cues = NULL)
```

Arguments

`data`	dataframe. A model training dataset. An m x n dataframe containing n cue values for each of the m exemplars.
`formula`	A formula
`tree.definitions`	dataframe. Definitions of one or more trees. The dataframe must contain the columns: cues, classes, thresholds, directions, exits.
`sens.w`	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. Only used for calculating wacc values.
`cost.outcomes`	numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
`cost.cues`	dataframe. A dataframe with two columns specifying the cost of each cue. The first column should be a vector of cue names, and the second column should be a numeric vector of costs. Cues in the dataset not present in `cost.cues` are assume to have 0 cost.

Value

A list of length 4 containing

auc — *Calculates AUC (Area under the Curve) using trapezoidal approximation*

Description

Calculates AUC (Area under the Curve) using trapezoidal approximation

Usage

```
auc(sens.v, spec.v)
```

Arguments

sens.v	a vector of sensitivities
spec.v	A vector of specificities

Examples

```
# Calculate the AUC for a vector of hit rates and false alarm rates

auc(sens.v = c(.1, .3, .5, .7), spec.v = c(.05, .1, .15, .3))
```

---

blood *Blood donation dataset*

Description

Blood donation dataset

Usage

```
blood
```

Format

A data frame containing 748 rows and 5 columns

recency Months since last donation

frequency Total number of donations

total Total blood donated in c.c.

time Months since first donation

donation.crit Did he/she donated blood in March 2007? ...

Source

https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center

breastcancer — *Dataset: Physiological dataset for 699 patients tested for breast cancer.*

Description

Dataset: Physiological dataset for 699 patients tested for breast cancer.

Usage

```
breastcancer
```

Format

A data frame containing 699 rows and 9 columns

thickness Clump Thickness

cellsize.unif Uniformity of Cell Size

cellshape.unif Uniformity of Cell Shape

adhesion Marginal Adhesion

epithelial Single Epithelial Cell Size

nuclei.bare Bare Nuclei

chromatin Bland Chromatin

nucleoli Normal Nucleoli

mitoses Mitoses

diagnosis Is cancer present? T or F

...

Source

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

car — *Car acceptability data*

Description

Car acceptability data

Usage

```
car
```

Format

A data frame containing 1728 rows and 7 columns

buying.price Numeric

maint.price Factor

doors Factor

persons Numeric

luggage Numeric

safety Factor

acceptability Factor

...

Source

http://archive.ics.uci.edu/ml/datasets/Car+Evaluation

`classtable` *Calculates several classification statistics from binary prediction and criterion (e.g.; truth) vectors*

Description

Calculates several classification statistics from binary prediction and criterion (e.g.; truth) vectors

Usage

```
classtable(prediction.v, criterion.v, sens.w = 0.5, cost.v = NULL,
  cost.outcomes = c(0, 1, 1, 0))
```

Arguments

`prediction.v`	A binary vector of predictions
`criterion.v`	A binary vector of criterion (true) values
`sens.w`	numeric. Weight given to sensitivity, must range from 0 to 1.
`cost.v`	numeric. An optional vector of additional costs to be added to each case.
`cost.outcomes`	numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.

Examples

```
 # classification statistics for 5 cases
classtable(prediction.v = c(0, 0, 0, 1, 1),
           criterion.v = c(0, 0, 1, 0, 1))
```

---

comp.pred *Wrapper for classfication algorithms*

Description

This function is a wrapper for many classification algorithms such as CART (rpart::rpart), logistic regression (glm), support vector machines (svm::svm) and random forests (randomForest::randomForest)

Usage

```
comp.pred(formula, data.train, data.test = NULL, algorithm = NULL,
  new.factors = "exclude")
```

Arguments

formula	a formula
data.train	dataframe. A training dataset
data.test	dataframe. A testing dataset
algorithm	string. An algorithm in the set "lr" – logistic regression, cart" – decision trees, "rlr" – regularised logistic regression, "svm" – support vector machines, "rf" – random forests
new.factors	string. What should be done if new factor values are discovered in the test set? "exclude" = exclude (i.e.; remove these cases), "base" = predict the base rate of the criterion.

Examples

```
# Fit many alternative algorithms to the mushrooms dataset


mushrooms.cart.pred <- comp.pred(formula = poisonous ~.,
                                 data.train = mushrooms[1:100,],
                                 data.test = mushrooms[101:nrow(mushrooms),],
                                 algorithm = "cart")

mushrooms.rf.pred <- comp.pred(formula = poisonous ~.,
```

```
                                data.train = mushrooms[1:100,],
                                data.test = mushrooms[101:nrow(mushrooms),],
                                algorithm = "rf")

mushrooms.svm.pred <- comp.pred(formula = poisonous ~.,
                                data.train = mushrooms[1:100,],
                                data.test = mushrooms[101:nrow(mushrooms),],
                                algorithm = "svm")
```

`contraceptive` *Contraceptive use data*

Description

Contraceptive use data

Usage

```
contraceptive
```

Format

A data frame containing 1473 rows and 10 columns

wife.age Numeric

wife.edu Factor

hus.ed Factor

children Numeric

wife.rel Numeric

wife.work Factor

hus.occ Factor

sol Factor

media Numeric

cont.crit numeric

...

Source

https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice

creditapproval *Credit approval data*

Description

Credit approval data

Usage

```
creditapproval
```

Format

A data frame containing 690 rows and 15 columns

Source

https://archive.ics.uci.edu/ml/datasets/Credit+Approval

cuerank *Calculates thresholds that maximize a statistic (goal) for cues.*

Description

Calculates thresholds that maximize a statistic (goal) for cues.

Usage

```
cuerank(formula = NULL, data = NULL, goal = "bacc", sens.w = 0.5,
  cost.outcomes = c(0, 1, 1, 0), cost.cues = NULL, numthresh.method = "o",
  rounding = NULL, factor.directions = c("=", "!="),
  numeric.directions = c(">", "<"), considerFALSE = TRUE,
  progress = FALSE, cue.rules = NULL)
```

Arguments

formula	formula. A formula specifying a binary criterion as a function of multiple variables
data	dataframe. A dataframe containing variables in formula
goal	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "wacc" = weighted accuracy, "d" = dprime
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity.

`cost.outcomes` numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.

`cost.cues` dataframe. A dataframe with two columns specifying the cost of each cue. The first column should be a vector of cue names, and the second column should be a numeric vector of costs. Cues in the dataset not present in `cost.cues` are assume to have 0 cost.

`numthresh.method` character. A string indicating how to calculate cue splitting thresholds. "m" = median split, "o" = split that maximizes the goal,

`rounding` integer. An integer indicating digit rounding for non-integer numeric cue thresholds. The default is NULL which means no rounding. A value of 0 rounds all possible thresholds to the nearest integer, 1 rounds to the nearest .1 (etc.).

`factor.directions` character. A vector of possible directions for factor values. `c("=", "!=")` allows both equality and inequality, while `"="` only allows for equality.

`numeric.directions` character. A vector of possible directions for numeric values. `c(">", "<")` allows only strict inequalities while `c("<=", "<", ">=", ">")` is more flexible.

`considerFALSE` logical. Should FALSE logical values be considered as potential thresholds? This is only relevant for very special algorithms.

`progress` logical. Should ongoing diagnostics be printed?

`cue.rules` dataframe. An optional df specifying existing cue thresholds, directions, names, and classes

Value

A dataframe containing thresholds and marginal classification statistics for each cue

Examples

```
## Not run:
 # What are the best thresholds for each cue in the mushrooms dataset?
 mushrooms.cues <- cuerank(formula = poisonous ~.,
                           data = mushrooms)

## End(Not run)
```

factclean — *Does miscellaneous cleaning of prediction datasets*

Description

Does miscellaneous cleaning of prediction datasets

Usage

```
factclean(data.train, data.test, show.warning = T)
```

Arguments

data.train	A training dataset
data.test	A testing dataset
show.warning	...

fan.algorithm — *Grows fast-and-frugal trees using the fan algorithm*

Description

Grows fast-and-frugal trees using the fan algorithm

Usage

```
fan.algorithm(formula, data, max.levels = 5, algorithm = "ifan",
  goal = "wacc", goal.chase = "bacc", sens.w = 0.5, cost.outcomes = c(0,
  1, 1, 0), cost.cues = NULL, numthresh.method = "o",
  stopping.rule = "exemplars", stopping.par = 0.1, rounding = NULL,
  progress = TRUE)
```

Arguments

formula	formula. A formula
data	dataframe. A dataset
max.levels	integer. The maximum number of levels in the tree(s)
algorithm	character. A string indicating how to rank cues during tree construction. "ifan" (independent fan) means that cues will only be ranked once with the entire training dataset "dfan" (dependent fan) means that cues will be ranked after each level in the tree with the remaining unclassified training exemplars.
goal	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy

goal.chase	character. A string indicating the statistic to maximize when constructing trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity.
cost.outcomes	numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
cost.cues	dataframe. A dataframe with two columns specifying the cost of each cue. The first column should be a vector of cue names, and the second column should be a numeric vector of costs. Cues in the dataset not present in `cost.cues` are assume to have 0 cost.
numthresh.method	character. How should thresholds for numeric cues be determined? `"o"` will optimize thresholds, while `"m"` will always use the median.
stopping.rule	character. A string indicating the method to stop growing trees. `"levels"` means the tree grows until a certain level, `"exemplars"` means the tree grows until a certain number of unclassified exemplars remain. `"statdelta"` means the tree grows until the change in the criterion statistic is less than a specified level.
stopping.par	numeric. A number indicating the parameter for the stopping rule. For stopping.rule == `"levels"`, this is the number of levels. For stopping rule `"exemplars"`, this is the smallest percentage of examplars allowed in the last level.
rounding	integer. How much should threshold parameters be rounded? Default is
progress	logical. Should tree growing progress be displayed?
...	Currently ignored

Value

A definition of an FFT

`fertility` *Fertility data set*

Description

Fertility data set

Usage

```
fertility
```

Format

A data frame containing 100 rows and 10 columns

Source

https://archive.ics.uci.edu/ml/datasets/Fertility

`FFForest` *Creates a forest of fast and frugal decision trees*

Description

This function is currently in development. The idea is to generate a random forest of fast and frugal trees from many splits of the training dataset.

Usage

```
FFForest(formula = NULL, data = NULL, data.test = NULL, max.levels = 5,
  ntree = 10, train.p = 0.5, algorithm = "ifan", goal = "wacc",
  goal.chase = "wacc", sens.w = 0.5, verbose = TRUE, cpus = 1,
  comp = FALSE, do.lr = TRUE, do.cart = TRUE, do.rf = TRUE,
  do.svm = TRUE, rank.method = NULL, hr.weight = NULL)
```

Arguments

`formula`	formula. A formula specifying a binary criterion as a function of multiple variables
`data`	dataframe. A dataframe containing variables in formula
`data.test`	dataframe. An optional dataframe of test data
`max.levels`	integer. Maximum number of levels considered for the trees.
`ntree`	integer. Number of trees to create.
`train.p`	numeric. What percentage of the data should be used to fit each tree? Smaller values will result in more diverse trees.
`algorithm`	string. The algorith uses to create FFTs. See arguments in `FFTrees()`
`goal`	character. A string indicating the statistic to maximize when selecting final trees: "acc" = overall accuracy, "bacc" = balanced accuracy, "d" = d-prime
`goal.chase`	character. A string indicating the statistic to maximize when constructing trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy
`sens.w`	numeric. How much weight to give to maximizing hits versus minimizing false alarms (between 0 and 1)
`verbose`	logical. Should progress reports be printed?
`cpus`	integer. Number of cpus to use. Any value larger than 1 will initiate parallel calculations in snowfall.
`comp, do.lr, do.cart, do.rf, do.svm`	logical. See arguments in `FFTrees()`
`rank.method, hr.weight`	depricated arguments

Value

An object of class `FFForest` with the following elements...

Examples

```
## Not run:
cancer.fff <- FFForest(formula = diagnosis ~.,
                       data = breastcancer,
                       ntree = 10,
                       train.p = .5,
                       cpus = 1)

## End(Not run)
```

`FFTrees` *Creates a fast-and-frugal trees (FFTrees) object.*

Description

This is the workhorse function for the `FFTrees` package. It creates (one or more) fast-and-frugal decision trees trained on a training dataset and tested on an optional test dataset.

Usage

```
FFTrees(formula = NULL, data = NULL, data.test = NULL,
  algorithm = "ifan", max.levels = NULL, sens.w = 0.5,
  cost.outcomes = NULL, cost.cues = NULL, stopping.rule = "exemplars",
  stopping.par = 0.1, goal = "wacc", goal.chase = "bacc",
  numthresh.method = "o", decision.labels = c("False", "True"),
  main = NULL, train.p = 1, rounding = NULL, progress = TRUE,
  my.tree = NULL, tree.definitions = NULL, comp = TRUE, do.cart = TRUE,
  do.lr = TRUE, do.rf = TRUE, do.svm = TRUE, store.data = FALSE,
  object = NULL, rank.method = NULL, force = FALSE, verbose = NULL)
```

Arguments

`formula`	formula. A formula specifying a logical criterion as a function of 1 or more predictors.
`data`	dataframe. A training dataset.
`data.test`	dataframe. An optional testing dataset with the same structure as data.
`algorithm`	character. The algorithm to create FFTs. Can be `'ifan'`, `'dfan'`, `'max'`, or `'zigzag'`.

`max.levels` integer. The maximum number of levels considered for the trees. Because all permutations of exit structures are considered, the larger `max.levels` is, the more trees will be created.

`sens.w` numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. Only relevant when `goal = 'wacc'`

`cost.outcomes` numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.

`cost.cues` dataframe. A dataframe with two columns specifying the cost of each cue. The first column should be a vector of cue names, and the second column should be a numeric vector of costs. Cues in the dataset not present in `cost.cues` are assume to have 0 cost.

`stopping.rule` character. A string indicating the method to stop growing trees. "levels" means the tree grows until a certain level. "exemplars" means the tree grows until a certain number of unclassified exemplars remain. "statdelta" means the tree grows until the change in the criterion statistic is less than a specified level.

`stopping.par` numeric. A number indicating the parameter for the stopping rule. For stopping.rule == "levels", this is the number of levels. For stopping rule == "exemplars", this is the smallest percentage of examplars allowed in the last level.

`goal` character. A string indicating the statistic to maximize when selecting final trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy

`goal.chase` character. A string indicating the statistic to maximize when constructing trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy

`numthresh.method` character. How should thresholds for numeric cues be determined? `"o"` will optimize thresholds, while `"m"` will always use the median.

`decision.labels` string. A vector of strings of length 2 indicating labels for negative and positive cases. E.g.; `decision.labels = c("Healthy", "Diseased")`

`main` string. An optional label for the dataset. Passed on to other functions like `plot.FFTrees()`, and `print.FFTrees()`

`train.p` numeric. What percentage of the data to use for training when `data.test` is not specified? For example, `train.p = .5` will randomly split `data` into a 50% training set and a 50% test set. `train.p = 1`, the default, uses all data for training.

`rounding` integer. An integer indicating digit rounding for non-integer numeric cue thresholds. The default is NULL which means no rounding. A value of 0 rounds all possible thresholds to the nearest integer, 1 rounds to the nearest .1 (etc.).

`progress` logical. Should progress reports be printed? Can be helpful for diagnosis when the function is running slowly.

`my.tree` string. A string representing an FFT in words. For example, `my.tree = "If age > 20, predict TRUE.`

`tree.definitions`
: dataframe. An optional hard-coded definition of trees (see details below). If specified, no new trees are created.

`comp, do.cart, do.lr, do.rf, do.svm`
: logical. Should alternative algorithms be created for comparison? cart = regular (non-frugal) trees with `rpart`, lr = logistic regression with `glm`, rf = random forests with `randomForest`, svm = support vector machines with `e1071`. Setting `comp = FALSE` sets all these arguments to FALSE.

`store.data`
: logical. Should training / test data be stored in the object? Default is FALSE.

`object`
: FFTrees. An optional existing FFTrees object. When specified, no new trees are fitted and the existing trees are applied to `data` and `data.test`.

`rank.method, verbose`
: depricated arguments.

`force`
: logical. If TRUE, forces some parameters (like goal) to be as specified by the user even when the algorithm thinks those specifications don't make sense.

Value

An `FFTrees` object with the following elements

formula The formula specified when creating the FFTs.

data.desc Descriptive statistics of the data

cue.accuracies Marginal accuracies of each cue given a decision threshold calculated with the specified algorithm

tree.definitions Definitions of each tree created by `FFTrees`. Each row corresponds to one tree. Different levels within a tree are separated by semi-colons. See above for more details.

tree.stats Tree definitions and classification statistics. Training and test data are stored separately

cost A list of cost information for each case in each tree.

level.stats Cumulative classification statistics at each tree level. Training and test data are stored separately

decision Final classification decisions. Each row is a case and each column is a tree. For example, row 1 in column 2 is the classification decision of tree number 2 for the first case. Training and test data are stored separately.

levelout The level at which each case is classified in each tree. Rows correspond to cases and columns correspond to trees. Training and test data are stored separately.

tree.max The index of the 'final' tree specified by the algorithm. For algorithms that only return a single tree, this value is always 1.

inwords A verbal definition of `tree.max`.

auc Area under the curve statistics

params A list of defined control parameters (e.g.; `algorithm`, `goal`)

comp Models and classification statistics for competitive classification algorithms: Regularized logistic regression, CART, and random forest.

data The original training and test data (only included when `store.data = TRUE`)

Examples

```
# Create ffts for heart disease
heart.fft <- FFTrees(formula = diagnosis ~.,
                     data = heartdisease)

# Print the result for summary info
heart.fft

# Plot the best tree
plot(heart.fft)
```

`FFTrees.guide` *Opens the FFTrees package guide*

Description

Opens the FFTrees package guide

Usage

```
FFTrees.guide()
```

`forestfires` *forestfires*

Description

A dataset of forest fire statistics.

Usage

```
forestfires
```

Format

A data frame containing 517 rows and 13 columns

X Integer -x-axis spatial coordinate within the Montesinho park map: 1 to 9

Y Integer - y-axis spatial coordinate within the Montesinho park map: 2 to 9

month Factor - month of the year: "jan" to "dec"

day Factor -day of the week: "mon" to "sun"

FFMC Numeric -FFMC index from the FWI system: 18.7 to 96.20

DMC Numeric - DMC index from the FWI system: 1.1 to 291.3

DC Numeric - DC index from the FWI system: 7.9 to 860.6

ISI Numeric - ISI index from the FWI system: 0.0 to 56.10

temp Numeric - temperature in Celsius degrees: 2.2 to 33.30

RH Numeric - relative humidity in percent: 15.0 to 100

wind Numeric - wind speed in km/h: 0.40 to 9.40

rain Numeric - outside rain in mm/m2 : 0.0 to 6.4

area Numeric - the burned area of the forest (in ha): 0.00 to 1090.84

...

Source

http://archive.ics.uci.edu/ml/datasets/Forest+Fires

`grow.FFTrees` *Grows fast-and-frugal trees using an algorithm specified by* `algorithm`.

Description

Grows fast-and-frugal trees using an algorithm specified by `algorithm`.

Usage

```
grow.FFTrees(formula, data, max.levels = NULL, algorithm = "ifan",
  goal = "wacc", goal.chase = "bacc", sens.w = 0.5, cost.outcomes = c(0,
  1, 1, 0), cost.cues = NULL, numthresh.method = "o",
  stopping.rule = "exemplars", stopping.par = 0.1, progress = FALSE,
  rank.method = NULL, cue.accuracies = NULL, ...)
```

Arguments

`formula`	formula. A formula
`data`	dataframe. A dataset
`max.levels`	integer. The maximum number of levels in the tree(s)
`algorithm`	character. A string indicating how to rank cues during tree construction. "m" (for ifan) means that cues will only be ranked once with the entire training dataset. "c" (conditional) means that cues will be ranked after each level in the tree with the remaining unclassified training exemplars.
`goal`	character. A string indicating the statistic to maximize: "acc" = overall accuracy, "bacc" = balanced accuracy, "wacc" = weighted accuracy
`goal.chase`	character. A string indicating the statistic to maximize when constructing trees: "acc" = overall accuracy, "wacc" = weighted accuracy, "bacc" = balanced accuracy

sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity.
cost.outcomes	numeric. A vector of length 4 specifying the costs of a hit, false alarm, miss, and correct rejection rspectively. E.g.; `cost.outcomes = c(0, 10, 20, 0)` means that a false alarm and miss cost 10 and 20 respectively while correct decisions have no cost.
cost.cues	dataframe. A dataframe with two columns specifying the cost of each cue. The first column should be a vector of cue names, and the second column should be a numeric vector of costs. Cues in the dataset not present in `cost.cues` are assume to have 0 cost.
numthresh.method	character. How should thresholds for numeric cues be determined? `"o"` will optimize thresholds, while `"m"` will always use the median.
stopping.rule	character. A string indicating the method to stop growing trees. "levels" means the tree grows until a certain level. "exemplars" means the tree grows until a certain number of unclassified exemplars remain. "statdelta" means the tree grows until the change in the criterion statistic is less than a specified level.
stopping.par	numeric. A number indicating the parameter for the stopping rule. For stopping.rule == "levels", this is the number of levels. For stopping rule == "exemplars", this is the smallest percentage of examplars allowed in the last level.
progress	logical. Should tree growing progress be displayed?
rank.method	depricated arguments
cue.accuracies	depricated arguments
...	Currently ignored

Value

A list of length 4. tree.definitions contains definitions of the tree(s). tree.stats contains classification statistics for the tree(s). levelout shows which level in the tree(s) each exemplar is classified. Finally, decision shows the classification decision for each tree for each exemplar

Examples

```
 titanic.trees <- grow.FFTrees(formula = survived ~.,
                               data = titanic)

# Tree definitions are stored in tree.definitions

titanic.trees$tree.definitions

# Tree classification statistics are in tree.stats

titanic.trees$tree.stats

# The level at which each exemplar is classified for each tree is in levelout

titanic.trees$levelout
```

```
# The decision for each exemplar for each tree is in decision

titanic.trees$decision
```

---

heart.cost *Cue costs for the heartdisease dataa*

Description

Cue costs for the heartdisease dataa

Usage

```
heart.cost
```

Format

A data frame containing 153 rows and 14 columns

cue The name of the cue

cost The cost of the cue

...

Source

https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/costs/

---

heart.fff *An FFForest object trained on the heartdisease dataset.*

Description

An FFForest object trained on the heartdisease dataset.

Usage

```
heart.fff
```

Format

An FFForest object

Details

The code used to generate this object was `heart.fff <- FFForest(diagnosis ~., data = heartdisease, ntree = 100`

`heart.test` *Heartdisease testing dataset*

Description

Testing data for a heart disease dataset. These data are used to test the prediction performance of a model trained on the `heart.train` data. The dataset `heartdisease` contains both datasets.

Usage

```
heart.test
```

Format

A data frame containing 153 rows and 14 columns

age Age

sex Sex, 1 = male, 0 = female

cp Chest pain type: ta = typical angina, aa = atypical angina, np = non-anginal pain, a = asymptomatic

trestbps Resting blood pressure (in mm Hg on admission to the hospital)

chol Serum cholestoral in mg/dl

fbs Fasting blood sugar > 120 mg/dl: 1 = true, 0 = false

restecg Resting electrocardiographic results. "normal" = normal, "abnormal" = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), "hypertrophy" = showing probable or definite left ventricular hypertrophy by Estes' criteria.

thalach Maximum heart rate achieved

exang Exercise induced angina: 1 = yes, 0 = no

oldpeak ST depression induced by exercise relative to rest

slope The slope of the peak exercise ST segment.

ca Number of major vessels (0-3) colored by flourosopy

thal "normal" = normal, "fd" = fixed defect, "rd" = reversable defect

diagnosis 1 = Heart disease, 0 = No Heart disease

...

Source

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

`heart.train` *Heartdisease training dataset.*

Description

Training data for a heart disease dataset used to train a model. The corresponding dataset for model testing is `heart.test`. The dataset `heartdisease` contains both datasets.

Usage

```
heart.train
```

Format

A data frame containing 150 rows and 14 columns

age Age

sex Sex, 1 = male, 0 = female

cp Chest pain type: ta = typical angina, aa = atypical angina, np = non-anginal pain, a = asymptomatic

trestbps Resting blood pressure (in mm Hg on admission to the hospital)

chol Serum cholestoral in mg/dl

fbs Fasting blood sugar > 120 mg/dl: 1 = true, 0 = false

restecg Resting electrocardiographic results. "normal" = normal, "abnormal" = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), "hypertrophy" = showing probable or definite left ventricular hypertrophy by Estes' criteria.

thalach Maximum heart rate achieved

exang Exercise induced angina: 1 = yes, 0 = no

oldpeak ST depression induced by exercise relative to rest

slope The slope of the peak exercise ST segment.

ca Number of major vessels (0-3) colored by flourosopy

thal "normal" = normal, "fd" = fixed defect, "rd" = reversable defect

diagnosis 1 = Heart disease, 0 = No Heart disease

...

Source

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

`heartdisease` *Heart disease dataset*

Description

Data for 303 patients tested for heart disease

Usage

```
heartdisease
```

Format

A data frame containing 303 rows and 14 columns

age Age

sex Sex, 1 = male, 0 = female

cp Chest pain type: ta = typical angina, aa = atypical angina, np = non-anginal pain, a = asymptomatic

trestbps Resting blood pressure (in mm Hg on admission to the hospital)

chol Serum cholestoral in mg/dl

fbs Fasting blood sugar > 120 mg/dl: 1 = true, 0 = false

restecg Resting electrocardiographic results. "normal" = normal, "abnormal" = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), "hypertrophy" = showing probable or definite left ventricular hypertrophy by Estes' criteria.

thalach Maximum heart rate achieved

exang Exercise induced angina: 1 = yes, 0 = no

oldpeak ST depression induced by exercise relative to rest

slope The slope of the peak exercise ST segment.

ca Number of major vessels (0-3) colored by flourosopy

thal "normal" = normal, "fd" = fixed defect, "rd" = reversable defect

diagnosis 1 = Heart disease, 0 = No Heart disease

...

Source

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

heuristic.algorithm *Heuristic algorithms for building FFTs originally developed by Martignon, Katsikopoulos & Woike (2008)*

Description

Heuristic algorithms for building FFTs originally developed by Martignon, Katsikopoulos & Woike (2008)

Usage

```
heuristic.algorithm(formula, data, max.levels = NULL, algorithm = "max")
```

Arguments

formula	formula. A formula specifying a logical criterion as a function of 1 or more predictors.
data	dataframe. A training dataset.
max.levels	integer. The maximum number of levels considered for the trees.
algorithm	string. Either 'max' or 'zigzag'

Value

A definition of an FFT

inwords *Describes an FFT in words*

Description

Describes an FFT in words

Usage

```
inwords(x = NULL, tree = NULL, classes.v = NULL, cues.v = NULL,
  directions.v = NULL, thresholds.v = NULL, exits.v = NULL,
  decision.labels = NULL, digits = 2)
```

Arguments

x	FFTrees. An FFTrees object created with FFTrees()
tree	integer. An integer specifying which tree in the object to verbalise. The default is x$tree.max
classes.v	integer.
cues.v	integer.
directions.v	string.
thresholds.v	string.
exits.v	numeric.
decision.labels	string. A string vector (of length 2) indicating labels for negative (0 or FALSE) and positive (1 or TRUE) cases in the crterion
digits	integer. How many digits to round numeric values

Value

A list of string vectors

Examples

```
heart.fft <- FFTrees(diagnosis ~.,
                     data = heartdisease,
                     decision.labels = c("Healthy", "Disease"))

inwords(heart.fft)
```

---

iris.v *Iris data set*

Description

Iris data set

Usage

```
iris.v
```

Format

A data frame containing 150 rows and 4 columns

Source

https://archive.ics.uci.edu/ml/datasets/Iris

`mushrooms` *Mushrooms dataset*

Description

Mushrooms dataset

Usage

```
mushrooms
```

Format

A data frame containing 8,124 rows and 23 columns (see http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.names for column descriptions)

poisonous numeric
cshape character
csurface character
ccolor character
bruises character
odor numeric
gattach character
gspace characte
gsize character
gcolor character
sshape character
sroot character
ssaring character
ssbring character
scaring character
scbring character
vtype character
vcolor character
ringnum character
ringtype character
sporepc character
population character
habitat character ...

Source

http://archive.ics.uci.edu/ml/datasets/Mushroom

plot.FFForest *Creates a network plot. Code taken from Dirk Wulff (www.dirkwulff.org)*

Description

Creates a network plot. Code taken from Dirk Wulff (www.dirkwulff.org)

Usage

```
## S3 method for class 'FFForest'
plot(x, node.cex.lim = c(1, 10), line.cex.lim = c(0.3,
  5), mincon = 0, lo = "kk", palette = NULL, ...)
```

Arguments

x	FFForest. An FFForest object created from FFForest()
node.cex.lim	numeric. Nodesize adjustment
line.cex.lim	numeric. Edgesize adjustment
mincon	integer. Minimum connection cutoff
lo	string. The layout of the network plot. Either 'kk' (Kamada-Kawai, the default), 'dh' (Davidson-Harel) or 'fr' (Fruchterman-Reingold)
palette	string. A string vector of colors
...	currently ignored

plot.FFTrees *Draws a FFTrees object.*

Description

The primary purpose of this function is to visualize a Fast and Frugal Tree (FFT) for data that has already been classified using the FFTrees() function. However, if the data have not yet been classified, the function can also implement a tree specified by the user. Inputs with the (M) header are manditory. If the tree has already been implimented, then only inputs with the (A) header should be entered. If the tree has not been implimented, then only inputs with the (B) header should be entered.

Usage

```
## S3 method for class 'FFTrees'
plot(x = NULL, data = "train", what = "tree",
  tree = "best.train", main = NULL, decision.labels = NULL,
  cue.cex = NULL, threshold.cex = NULL, decision.cex = 1, comp = TRUE,
  stats = TRUE, n.per.icon = NULL, which.tree = NULL,
  level.type = "bar", decision.names = NULL, ...)
```

Arguments

x	A FFTrees object created from "FFTrees()"
data	Either a dataframe of new data, or one of two strings 'train' or 'test'. In this case, the corresponding dataset in the x object will be used.
what	string. What should be plotted? 'tree' (the default) shows one tree (specified by 'tree'). 'cues' shows the marginal accuracy of cues in an ROC space.
tree	integer. An integer indicating which tree to plot (only valid when the tree argument is non-empty). To plot the best training (or test) tree with respect to the goal specified during FFT construction, use "best.train" or "best.test"
main	character. The main plot label.
decision.labels	character. A string vector of length 2 indicating the content-specific name for noise and signal cases.
cue.cex	numeric. The size of the cue labels.
threshold.cex	numeric. The size of the threshold labels.
decision.cex	numeric. The size of the decision labels.
comp	logical. Should the performance of competitive algorithms (e.g.; logistic regression, random forests etc.) be shown in the ROC plot (if available?)
stats	logical. Should statistical information be plotted? If FALSE, then only the tree (without any reference to statistics) will be plotted.
n.per.icon	Number of cases per icon
which.tree	deprecated argument, only for backwards compatibility, use "tree" instead.
level.type	string. How should bottom levels be drawn? Can be "bar" or "line"
decision.names	depricated arguments.
...	Currently ignored.

Examples

```
# Create FFTrees of the heart disease data
heart.fft <- FFTrees(formula = diagnosis ~.,
data = heartdisease)

# Visualise the tree
plot(heart.fft,
     main = "Heart Disease Diagnosis",
     decision.labels = c("Absent", "Present"))


# See the vignette for more details
vignette("FFTrees_plot", package = "FFTrees")
```

predict.FFForest *Predict outcoms from a test dataset using an FFForest object*

Description

Predict outcoms from a test dataset using an FFForest object

Usage

```
## S3 method for class 'FFForest'
predict(object = NULL, data = NULL, threshold = 0.5,
  ...)
```

Arguments

object	FFForest. An FFForest object created from the FFForest() function.
data	dataframe. A dataframe of test data
threshold	numeric. A threshold value
...	Additional arguments passed on to predict()

Value

A logical vector of predictions

predict.FFTrees *Predict new data from an FFTrees x*

Description

Predict new data from an FFTrees x

Usage

```
## S3 method for class 'FFTrees'
predict(object = NULL, data = NULL, tree = "best.train",
  sens.w = NULL, ...)
```

Arguments

object	An FFTrees object created from the FFTrees() function.
data	dataframe. A dataframe of test data
tree	Which tree in the FFTrees x should be used? Can be an integer or "best.train" (the default) to use the tree with the best training statistics (according the goal specified in tree construction).
sens.w	numeric. A number from 0 to 1 indicating how to weight sensitivity relative to specificity. If specified, the tree with the highest weighted accuracy (wacc) given the specified value of sens.w will be selected
...	Additional arguments passed on to predict()

Value

A logical vector of predictions

Examples

```
# Create training and test data

set.seed(100)
breastcancer <- breastcancer[sample(nrow(breastcancer)),]
breast.train <- breastcancer[1:150,]
breast.test <- breastcancer[151:303,]

# Create an FFTrees x from the training data

breast.fft <- FFTrees(formula = diagnosis ~.,
                              data = breast.train)

# Predict results for test data
breast.fft.pred <- predict(breast.fft,
                           data = breast.test)
```

---

print.FFForest *Prints summary information from an FFForest x*

Description

Printing function for an FFTrees x

Usage

```
## S3 method for class 'FFForest'
print(x = NULL, ...)
```

Arguments

x	FFForest. An FFForest x created from FFForest()
...	additional arguments passed to print.

---

print.FFTrees *Prints summary information from an FFTrees object*

Description

Printing function for an FFTrees object

Usage

```
## S3 method for class 'FFTrees'
print(x = NULL, ...)
```

Arguments

x	FFTrees. A FFTrees x created from FFTrees()
...	additional arguments passed to print.

---

showcues *Visualizes cue accuracies from an FFTrees object in a ROC space*

Description

Visualizes cue accuracies from an FFTrees object in a ROC space

Usage

```
showcues(x = NULL, data = "train", cue.accuracies = NULL, main = NULL,
  top = 5)
```

Arguments

x	An FFTrees object
data	A string indicating whether or not to show training ("train") or testing ("test") cue accuracies
cue.accuracies	dataframe. An optional dataframe specifying cue accuracies directly (without specifying an FFTrees object x)
main	Main plot description
top	An integer indicating how many of the top cues to highlight

sonar *Sonar data set*

Description

Sonar data set

Usage

```
sonar
```

Format

A data frame containing 208 rows and 60 columns

Source

https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)

summary.FFTrees *Returns a summary of an fft object*

Description

Returns a summary of an fft object

Usage

```
## S3 method for class 'FFTrees'
summary(object, ...)
```

Arguments

object	An FFTrees object
...	Additional arguments (currently ignored)

titanic *Titanic dataset*

Description

A dataset indicating who survived on the Titanic

Usage

```
titanic
```

Format

A data frame containing 2,201 rows and 4 columns

class Factor - Class (first, second, third, or crew)

age Factor - Age group (child or adult)

sex Factor - Sex (male or female)

survived Factor - Whether the passenger survived (1) or not (0)

...

Source

https://www.amstat.org/publications/jse/datasets/titanic.dat.txt

voting *Voting data set*

Description

Voting data set

Usage

```
voting
```

Format

A data frame containing 435 rows and 16 columns

Source

https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

wine *Wine tasting dataset*

Description

Chemical and tasting data from wines in North Portugal

Usage

```
wine
```

Format

A data frame containing 6497 rows and 13 columns

Source

http://archive.ics.uci.edu/ml/datasets/Wine+Quality

wordstoFFT *Converts text describing an FFT into an FFT definition.*

Description

Converts text describing an FFT into an FFT definition.

Usage

```
wordstoFFT(input, cue.names, decision.labels = NULL)
```

Arguments

input	string. A string describing an FFT in words (see examples)
cue.names	string. A vector of cue names
decision.labels	string. A vector of decision labels

Examples

```
my.tree.def <- wordstoFFT(input = "If age > 55, predict True.
                                If cp = {a,b,np}, predict False, otherwise, predict True",
                          cue.names = names(heartdisease))
```

Index