Package ‘glmtree’

October 6, 2019

Type Package

Title Logistic Regression Trees

Version 0.1

Date 2019-09-26

Maintainer Adrien Ehrhardt <adrien.ehrhardt@centraliens-lille.org>

Description A logistic regression tree is a decision tree with logistic regressions at its leaves. A particular stochastic expectation maximization algorithm is used to draw a few good trees, that are then assessed via the user's criterion of choice among BIC / AIC / test set Gini. The formal development is given in a PhD chapter, see Ehrhardt (2019) <https://github.com/adimajo/manuscrit_these/releases/>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports partykit, magrittr, methods, dplyr, caret

Suggests FactoMineR, knitr, testthat, covr, rmarkdown

URL https://adimajo.github.io

BugReports https://github.com/adimajo/glmtree/issues

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Adrien Ehrhardt [aut, cre]

Repository CRAN

Date/Publication 2019-10-06 12:50:02 UTC

R topics documented:

`generateData` *Generates data from two logistic regression trees.*

Description

This function generates data from two logistic regression trees: one with three apparent clusters (in terms of variance of the features) but a single logistic regression generating y | x, and one with a single apparent cluster but three different logistic regressions generating y | x given a categorical feature.

Usage

```
generateData(n = 100, scenario = "tree", visualize = FALSE)
```

Arguments

`n` The number of observations to draw.

`scenario` The "no tree" scenario denotes the first scenario where there is a single logistic regression generating the data. The "tree" scenario generates data from the second data generating mechanism where there are three logistic regressions.

`visualize` Whether (TRUE) or not (FALSE) to plot the generated data.

Value

Generates and returns data according to a true logistic regression tree (if scenario = "tree") or a single regression tree (if scenario = "no tree"). Eventually plots this dataset (if visualize = TRUE).

Author(s)

Adrien Ehrhardt

Examples

```
generateData(scenario = "tree")
```

glmdisc-class *Class glmtree*

Description

Class glmtree represents a logistic regression tree scheme associated with its optimal logistic regression models.

Slots

parameters The parameters associated with the method.

best.tree The best discretization scheme found by the method given its parameters.

performance The performance obtained with the method given its parameters.

glmtree *Logistic regression tree by Stochastic-Expectation-Maximization*

Description

This function produces a logistic regression tree: a decision tree with logistic regressions at its leaves.

Usage

```
glmtree(x, y, K = 10, iterations = 200, test = FALSE,
  validation = FALSE, proportions = c(0.3), criterion = "bic",
  ctree_controls = partykit::ctree_control(alpha = 0.1, minbucket = 100,
  maxdepth = 5))
```

Arguments

x The features to use for prediction.

y The binary / boolean labels to predict.

K The number of segments to start with (maximum number of segments there'll be in the end).

iterations The number of iterations to do in the SEM protocole (default: 200).

test Boolean : True if the algorithm should use predictors to construct a test set on which to calculate the provided criterion using the best discretization scheme (chosen thanks to the provided criterion on either the test set (if true) or the training set (otherwise)) (default: TRUE).

validation Boolean : True if the algorithm should use predictors to construct a validation set on which to search for the best discretization scheme using the provided criterion (default: TRUE).

`proportions` The list of the proportions wanted for test and validation set. Not used when both test and validation are false. Only the first is used when there is only one of either test or validation that is set to TRUE. Produces an error when the sum is greater to one. Default: list(0.2,0.2) so that the training set has 0.6 of the input observations.

`criterion` The criterion ('gini','aic','bic') to use to choose the best discretization scheme among the generated ones (default: 'gini'). Nota Bene: it is best to use 'gini' only when test is set to TRUE and 'aic' or 'bic' when it is not. When using 'aic' or 'bic' with a test set, the likelihood is returned as there is no need to penalize for generalization purposes.

`ctree_controls` The controls to use for 'partykit::ctree'.

Value

An S4 object of class 'glmtree' that contains the parameters used to search for the logistic regression tree, the best tree it found, and its performance.

Author(s)

Adrien Ehrhardt

Examples

```
data = generateData(n = 100, scenario = "no tree")
glmtree(x = data[,c("x1", "x2")], y = data$y, K = 5, iterations = 80, criterion = "aic")
```

`normalizedGini` *Calculating the normalized Gini index*

Description

This function calculates the Gini index of a classification rule outputting probabilities. It is a classical metric in the context of Credit Scoring. It is equal to 2 times the AUC (Area Under ROC Curve) minus 1.

Usage

```
normalizedGini(actual, predicted)
```

Arguments

`actual` The numeric binary vector of the actual labels observed.

`predicted` The vector of the probabilities predicted by the classification rule.

Value

The Gini index of the predicted probabilities as a single 'num'.

Author(s)

Adrien Ehrhardt

Examples

```
normalizedGini(c(1,1,1,0,0),c(0.7,0.9,0.5,0.6,0.3))
```

Index