

Package ‘CaseBasedReasoning’

June 12, 2018

Type Package

Title Case-Based Reasoning

Version 0.1

Date 2018-06-06

Author Dr. Simon Mueller <simon.mueller@muon-stat.com>, PD Dr. Juergen Dippon <juergen.dippon@mathematik.uni-stuttgart.de>

Maintainer Dr. Simon Mueller <simon.mueller@muon-stat.com>

Description Given a large set of problems and their individual solutions case based reasoning seeks to solve a new problem by referring to the solution of that problem which is “most similar” to the new problem. Crucial in case based reasoning is the decision which problem “most closely” matches a given new problem. The basic idea is to define a family of distance functions and to use these distance functions as parameters of local averaging regression estimates of the final result. Then that distance function is chosen for which the resulting estimate is optimal with respect to a certain error measure used in regression estimation. The idea is based on: Dippon J. et al. (2002) <DOI:10.1016/S0167-9473(02)00058-0>.

BugReports <https://github.com/sipemu/case-based-reasoning/issues>

License AGPL

LazyData TRUE

NeedsCompilation yes

Imports R6, ranger, survival, tidyverse, cowplot, dplyr, data.table, magrittr, rms, Rcpp, RcppParallel

Suggests testthat, knitr, rmarkdown, RcppArmadillo

LinkingTo Rcpp, RcppArmadillo, RcppParallel

SystemRequirements C++11

LazyLoad yes

ByteCompile yes

VignetteBuilder knitr

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-06-12 10:34:11 UTC

R topics documented:

asDistObject	2
CaseBasedReasoning	2
CBRBase	3
CoxBetaModel	3
depthMatrix	4
distanceRandomForest	4
distanceTerminalNodes	5
forestToMatrix	6
proximityMatrix	6
RFModel	7
terminalNodeIDs	8
Validate	8
weightedDistance	9

Index	10
--------------	-----------

asDistObject	<i>Converts a distance vector into an object of class dist</i>
--------------	--

Description

Converts a distance vector into an object of class dist

Usage

```
asDistObject(x, n, method)
```

Arguments

x	data vector
n	length of x
method	method description

CaseBasedReasoning	<i>Case Based Reasoning</i>
--------------------	-----------------------------

Description

Case Based Reasoning

CBRBase	<i>Root class for common functionality of this package</i>
---------	--

Description

Root class for common functionality of this package

Usage

CBRBase

Format

An object of class R6ClassGenerator of length 24.

CoxBetaModel	<i>Cox-Beta Model</i>
--------------	-----------------------

Description

Regression beta coefficients are use for building a weighted distance measure between the learning and verum data set. The learning data set is used for learning the Cox model and use the obtained weights for calculating a (n x m)-distance matrix, where n is the number of cases in the learning data set and m is the number of cases of the query data. This distance matrix can then be used for cluster analysis or for getting for each case in the query data k (=1,...,l) smilar cases from the learning data. The rms-package is used for model fitting, variable selection, and checking the assumptions. If query data is ommitted, a n x n- distance matrix is returned.

Usage

CoxBetaModel

Format

An [R6Class](#) generator object

Usage

For usage details see **Methods, Arguments, and Examples** sections.

Methods

`new(formula, ...)` This method is used to create an object of this class R6Model. Formula for analysis has to be provided. Further parameters for the ranger call can be provided here.

`fit(dtData)` Fits the RandomForest model.

... See [CBRBase](#) class.

depthMatrix *Get depth distance matrix*

Description

Get depth distance matrix

Usage

```
depthMatrix(x, y = NULL, rfObject)
```

Arguments

x	a new dataset
y	a new dataset
rfObject	ranger object

Examples

```
require(ranger)
rf <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
d <- depthMatrix(x=iris[, -5], rfObject=rf)
```

distanceRandomForest *Distance calculation based on RandomForest Proximity or Depth*

Description

Distance calculation based on RandomForest Proximity or Depth

Usage

```
distanceRandomForest(x, y = NULL, rfObject, method = "Proximity",
  threads = NULL)
```

Arguments

x	a data.frame
y	a second data.frame
rfObject	ranger object
method	distance calculation method
threads	number of threads to use

Value

a dist or a matrix object with pairwise distance of observations in x vs y (if not null)

Examples

```
library(ranger)
# proximity pairwise distances
rf.fit <- ranger(Species ~ ., data = iris, num.trees = 500, write.forest = TRUE)
distanceRandomForest(x = iris[, -5], rfObject = rf.fit, method = "Proximity", threads = 1)

# depth distance for train versus test subset
set.seed(1234L)
learn <- sample(1:150, 100)
test <- (1:150)[-learn]
rf.fit <- ranger(Species ~ ., data = iris[learn, ], num.trees = 500, write.forest = TRUE)
distanceRandomForest(x = iris[learn, -5], y = iris[test, -5], rfObject = rf.fit, method = "Depth")
```

distanceTerminalNodes *Calculate terminal node distance for each tree and terminal*

Description

first two columns are terminal node IDs; If an ID pair do not appear in a tree -1 is inserted

Usage

```
distanceTerminalNodes(rfObject)
```

Arguments

```
rfObject      ranger object
```

Value

a matrix object with pairwise terminal node edge length

Examples

```
require(ranger)
rf.fit <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
d <- distanceTerminalNodes(rf.fit)
```

forestToMatrix	<i>Forest2Matrix</i>
----------------	----------------------

Description

Transform trees of a ranger-object to a matrix

Usage

```
forestToMatrix(rfObject)
```

Arguments

rfObject ranger object

Value

a matrix object with Column 1: tree ID Column 2: node ID Column 3: child node ID 1 Column 4: child node ID 2

Examples

```
library(ranger)
rf.fit <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
forestMat <- forestToMatrix(rf.fit)
```

proximityMatrix	<i>Get proximity matrix of an ranger object</i>
-----------------	---

Description

Get proximity matrix of an ranger object

Usage

```
proximityMatrix(x, y = NULL, rfObject)
```

Arguments

x a new dataset
y a second new dataset (Default: NULL)
rfObject ranger object

Value

a dist or a matrix object with pairwise proximity of observations in x vs y (if not null)

Examples

```
require(ranger)
rf <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
d <- proximityMatrix(x = iris[, -5], rfObject = rf)

set.seed(1234L)
learn <- sample(1:150, 100)
test <- (1:150)[-learn]
rf <- ranger(Species ~ ., data = iris[learn, ], num.trees = 500, write.forest = TRUE)
d <- proximityMatrix(x = iris[learn, -5], y = iris[test, -5], rfObject = rf)
```

RFModel

RandomForest Proximity

Description

This class uses the proximity matrix of the random survival forest algorithm as a similarity matrix ($\sqrt{1 - \text{proximity matrix}}$) of learning and verum cases. By default all cases with at least one missing values are dropped from learning, calculating the distance matrix, and searching for similar cases.

Usage

RFModel

Format

[R6Class](#) object

Usage

For usage details see **Methods, Arguments, and Examples** sections.

Methods

`new(formula, ...)` This method is used to create an object of this class RFModel. Formula for analysis has to be provided. Further parameters for the ranger call can be provided here.

`fit(dtData)` Fits the RandomForest model.

`set_dist(method)` Set the proximity measure, 'Depth' (Default) and 'Proximity' are allowed.

... See [CBRBase](#) class.

References

Englund and Verikas. A novel approach to estimate proximity in a random forest: An exploratory study.

terminalNodeIDs	<i>Get the terminal node id of a RandomForest Object</i>
-----------------	--

Description

Extracts for each observation and for each tree in the forest the terminal node id.

Usage

```
terminalNodeIDs(x, rfObject)
```

Arguments

x	a data.frame
rfObject	ranger object

Value

Matrix with terminal node IDs for all observations in x (rows) and trees (columns)

Examples

```
library(ranger)
rf.fit <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
dfNodes <- terminalNodeIDs(iris[, -5], rf.fit)
```

Validate	<i>R6 Validation Class for case based reasoning</i>
----------	---

Description

R6 Validation Class for case based reasoning

Usage

```
Validate
```

Format

An [R6Class](#) generator object

weightedDistance	<i>Weighted Distance calculation</i>
------------------	--------------------------------------

Description

Weighted Distance calculation

Usage

```
weightedDistance(x, y = NULL, weights = NULL)
```

Arguments

x	a new dataset
y	a second new dataset
weights	a vector of weights

Value

a dist or matrix object

Examples

```
require(ranger)
rf <- ranger(Species ~ ., data = iris, num.trees = 5, write.forest = TRUE)
nodeID <- terminalNodeIDs(iris[, -5], rf)
```

Index

- *Topic **c++**
 - CBRBase, 3
- *Topic **data-preparation**
 - CBRBase, 3
- *Topic **datasets**
 - CoxBetaModel, 3
 - RFModel, 7
- *Topic **data**
 - CBRBase, 3
- *Topic **for**
 - CBRBase, 3
- *Topic **functions**
 - CBRBase, 3
- *Topic **integer**
 - CBRBase, 3
- *Topic **necessary**
 - CBRBase, 3
- *Topic **representation;**
 - CBRBase, 3
- *Topic **to**
 - CBRBase, 3
- *Topic **transforms**
 - CBRBase, 3
- *Topic **validation**
 - Validate, 8

asDistObject, 2

CaseBasedReasoning, 2

CaseBasedReasoning-package
(CaseBasedReasoning), 2

CBRBase, 3, 3, 7

CoxBetaModel, 3

depthMatrix, 4

distanceRandomForest, 4

distanceTerminalNodes, 5

forestToMatrix, 6

proximityMatrix, 6

R6Class, 3, 7, 8

RFModel, 7

terminalNodeIDs, 8

Validate, 8

weightedDistance, 9