Package 'FLR'

January 20, 2025

Title Fuzzy Logic Rule Classifier

Version 1.0

Date 2012-03-11

Author Constantinos Mavridis and Ioannis N. Athanasiadis

Maintainer Constantinos Mavridis <consmavr@gmail.com>

Description FLR algorithm for classification

License GPL (>= 2)

Repository CRAN

Date/Publication 2014-05-06 13:19:42

Depends combinat

NeedsCompilation no

Contents

accIs *Accuracy of FLR*

Description

Accuracy of the flr classification.

Usage

```
accIs(testData,testDataB)
```

Arguments

testData	an input data.frame of the test after classification.
testDataB	an input data.frame of the original test data.

Value

return the accuracy of the classification

dataset001 *dataset001*

Description

Dataset with 296 instances if 25 attributes.

Usage

```
data(dataset001)
```

Format

A data frame with 296 instances on the following 25 variables.

`state` 9 US states.

`county` County.

`site.id` Site id.

`latitude` Latitude.

`longitude` Longtitude.

`X2009.2011.dv` 2009.2011 dv.

`X2010.2012.dv` 2010.2012 dv.

`X2009.2011.design.value..ppm.2.3` 2009-2011 design value (ppm)2,3

`X2010.2012.design.value..ppm...estimated.` 2010-2012 design value (ppm) [estimated].

`X2009.2011.design.value.status4` 2009-2011 design value status4.

`percent.complete.in.20095` percent complete in 20095.

`percent.complete.in.20105` percent complete in 20105.

`percent.complete.in.20115` percent complete in 20115.

`X2009.2011.average.percent.complete` 2009-2011 average percent complete.

`X..of.days.above.the.naaqs.in.2009` # of days above the naaqs in 2009.

`X..of.days.above.the.naaqs.in.2010` # of days above the naaqs in 2010.

`X..of.days.above.the.naaqs.in.2011` # of days above the naaqs in 2011.

`X..of.days.above.the.naaqs.in.2012` # of days above the naaqs in 2012.

`X4th.highest.daily.max.value.in.2009` 4th highest daily max value in 2009.

`X4th.highest.daily.max.value.in.2010` 4th highest daily max value in 2010.

`X4th.highest.daily.max.value.in.2011` 4th highest daily max value in 2011.

`X4th.highest.daily.max.value.in.2012.` 4th highest daily max value in 2012.

`column_27` Column_27.

`column_29` Column_29.

`class` Class category.

Source

geocommons.com

References

geocommons.com

`denormDatal` *Denormalize Fuzzy Lattices.*

Description

Denormalize fuzzy lattices.

Usage

```
denormDatal(fuzlat,bounds)
```

Arguments

`fuzlat`	a fuzzy lattice containing mix and max value for each instance of the data set at the first columns,from left to right, followed by className and categ.
`bounds`	a 2 comumn matrix containing min and max value for each instance of the dataset.

Value

return denormalized fuzzy lattice.

fuzzyLatticec *Constructs A Fuzzy Lattice*

Description

Constructs a Fuzzy Lattice from an instance of the dataset.

Usage

```
fuzzyLatticec(dF,dR,bounds)
```

Arguments

dF	an empty list containing just the names for each fuzzy lattice column.
dR	an instance of the dataset
bounds	a 2 comumn matrix containing min and max value for each instance of the dataset.

Value

return a fuzzy lattice (min and max value for each attribute, className,categ).

indexCalc *Index Calculator*

Description

Returns a vector that contains the number of rules created for each class.

Usage

```
indexCalc(learnedCode)
```

Arguments

learnedCode	a data.frame of fuzzy lattices. Each lattice is a rule created with the trainNow function.

Value

return a vector that contains the number of rules created for each class.

mat *Graph distance matrix*

Description

A matrix containing the distances of the nodes in a graph.

Usage

```
data(mat)
```

Format

A data frame of 9 rows and 9 columns.

`Illinois` number
`Indiana` number
`Kentucky` number
`Michigan` number
`North.Carolina` number
`Ohio` number
`Pennsylvania` number
`Tennessee` number
`Virginia` number

normData *Normalize Data and Denormalize data.*

Description

Normalize Data to be in range of 0~1.

Usage

```
normData(data1)
denormData(data1,bounds)
```

Arguments

`data1`	an input data.frame where last instance must be the class instance and be named 'class'.
`bounds`	a 2 comumn matrix containing min and max value for each instance of the dataset.

Value

return normalized or denormalized data.frame.

prepData *Prepare Dataset*

Description

Alters the dataset in a form that can be used for training and classification.

Usage

```
prepData(data)
```

Arguments

data	an input data.frame where last instance must be the class instance and be named 'class'.

Value

return the data.frame without missing class instances and converts nominal attributes into numeric.

sepFlag *Flags Instances*

Description

Randomly flags instances in order to be used as training(0) or testing(1) data with the ratio depending on variable gg.

Usage

```
sepFlag(gg,data1)
```

Arguments

gg	percentage of instances to be used as training data for the classification.
data1	an input data.frame where last instance must be the class instance and be named 'class'.

Value

return original data with a flag column added at the end.

set_bounds *Creates A Boundaries File.*

Description

Creates a boundaries of min and max columns for each attribute of a dataset.

Usage

```
set_bounds(data1)
```

Arguments

data1	an input data.frame where last instance must be the class instance and be named 'class'.

Value

return a data.frame of 2 columns (min,max) for each instance of the data(NOT class).

spatdt *Spatial Data Handling*

Description

Creates a linear connection between spatial data in order to be used for classification.

Usage

```
spatdt(data,idx,mat,pre_order=0,snd=0)
get.cost(zzz,mat)
get.cost2(pre_order,mat)
get.pos(instz)
winner.route(cost)
```

Arguments

data	an input data.frame
idx	indicates the position of the spatial data attribute.
mat	a matrix indicating distances
pre_order	predefined order
snd	indicates which node will be used as the starting one. The default value 0 means that the best route will be chosen, without taking into consideration which the starting node will be.
zzz	a route
instz	instance
cost	cost of routes

Value

return a list of 3 objects: a) The modified dataset, b) winner route, c) the total distance of the route.

Examples

```
#Import data
data(dataset001)
data<-dataset001
data(mat)

idx<-1
rhoa<-0.6
param<-"sigmoid"
pre_order<-c(1,2,3,4,5,6,7,8,9)

#Data preprocess
data<-spatdt(data,idx,mat,pre_order)
```

`testD` *Creates Testing And Training Samples*

Description

Creates testing and training samples from the original data.

Usage

```
testD(data2)
trainD(data2)
```

Arguments

`data2`	a data.frame flaged with the sepFlag function.

Value

return the training and testing samples that will be used for the classification.

testNow *Testing Phase Of FLR*

Description

Implements classification using FLR on a data.frame.

Usage

```
testNow(testData,learnedCode)
```

Arguments

testData	an input data.frame.
learnedCode	a data.frame of fuzzy lattices. Each lattice is a rule created with the trainNow function.

Value

return the testData data.frame after classification.

trainNow *Training Phase Of FLR*

Description

Creates rules for classification using FLR.

Usage

```
trainNow(trainData,param,rhoa=0.5,l=6,x0=0.5,EPSILON=10^(-6))
join(inpBuf,num)
theta(x,x0,param)
ufun(x,x0,l,param)
valuation(fuzlat,x0,l,param)
createNframe(trainData)
createNlist(trainData)
```

Arguments

trainData	an input data.frame.
param	parameter indicating linear positive valuation for 0 and sigmoid positive valuation for 1. The default value is set to 0.
rhoa	vigilance parameter in range [0,1]. The default value is set to 0.6.
l	parameter of u and theta functions of FLR. The default value is set to 6.

x0	parameter of u and theta functions of FLR. The default value is set to 0.4.
EPSILON	parameter EPSILON.The default value is set to 10^(-6).
inpBuf	input buffer.
num	num
x	fuzzy lattice
fuzlat	fuzzy lattice

Value

return a data.frame of the learned code.

Index