Package 'dprep'

November 24, 2015

Type Package

Title Data Pre-Processing and Visualization Functions for Classification

Version 3.0.2

Date 2015-11-14

Author Edgar Acuna and the CASTLE research group at The University of Puerto Rico-Mayaguez

Maintainer Edgar Acuna <edgar.acuna@upr.edu>

Description
Data preprocessing techniques for classification. Functions for normalization, handling of missing values,discretization, outlier detection, feature selection, and data visualization are included.

Depends R (>= 3.1.0), graphics, stats

Imports MASS, e1071, class, nnet, rpart, FNN, StatMatch, rgl, methods

License GPL

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-11-24 07:46:38

R topics documented:

`dprep-package` *Data Preprocessing for supervised classification*

Description

Functions for normalization, treatment of missing values,discretization, outlier detection, feature selection, and visualization

Details

Dprep has been developed by Professor Edgar Acuna and his students at the CASTLE research Group of the University of Puerto Rico-Mayaguez. This is a library of R functions for normalization, handling of missing values, discretization, outlier detection, feature selection, and data visualization classification. Most of the methods handle datasets with numerical and categorical attributes.

Normalization methods: Score Normalization, Min-Max Normalization, Decimal scale, Sigmoidal Normalization, Softmax normalization.

Missing values methods: Imputation by mean, median and mode (categorical features), K-nn Imputation (categorical and numerical data).

Discretization Methods: Equal width bins, Equal Frequency bins, Holte's One R, chiMerge, Entropy Discretization with MDL stopping rule.

Feature Selection Methods: ReliefF, LVF, Finco, Sequential Forward Selection, Sequential Floating Forward Selection.

Outlier Detection Methods: Mahaout, Robout, Bay's algorthm, LOF.

Crossvalidation estimation error: LDA, Naive Bayes, Logistic, Knn, Rpart, Neural Networks.

Author(s)

Maintainer: Edgar Acuna <edgar.acuna@upr.edu>

References

See website academic.uprm.edu/eacuna/dprep.html

`acugow` *Gower distance from a vector to a matrix*

Description

This function finds out the gower distance between a vector and a matrix

Usage

```
acugow(x, data, vnom = NULL)
```

Arguments

`x`	A Vector of attributes values
`data`	A matrix dataset
`vnom`	A vector indicating the columns with nominal attributes in the matrix dataset

Value

matdist	a matrix containing the components of the distance vector from x to each row of data
dist	a vector containing the distances from x to each row of data

Author(s)

Edgar Acuna

See Also

reliefcont

Examples

```
data(crx)
crx.imp=ce.impute(crx,"knn",nomatr=c(1,4,5,6,7,9,10,12,13),3)
acugow(crx.imp[1,-16],crx.imp[-1,-16],vnom=c(1,4,5,6,7,9,10,12,13))
```

---

arboleje *Predicting a bank's decision to give a loan for buying a car.*

Description

Simulated example about predicting a bank's decision to give a loan to customer for buying a car.

Usage

```
data("arboleje")
```

Format

A data frame with 25 observations on the following 7 variables.

Sexo a factor indicaing the customer's gender with levels Hombre Mujer

Familia a numeric vector indicating the number of members in the family

CasaPropia a factor with levels No Si

AnosEmpleo a numeric vector indicating the years of employment

Sueldo a numeric vector indicating the monthly salary

StatusMarital a factor with levels Casado Divorciado Soltero Viudo

Prestamo a factor indicating the bank's with levels No Si

Source

Originated by EDgar Acuna

Examples

```
data(arboleje)
library(rpart)
rpart(Prestamo~.,data=arboleje,method="class")
```

---

arboleje1 *Predicting a bank's decision to give a loan for buying a car.*

Description

Simulated example about predicting a bank's decision to give a loan to customer for buying a car. The feature "Marital Status" has been codified using three dummy variables.

Usage

```
data("arboleje1")
```

Format

A data frame with 25 observations on the following 9 variables.

`Sexo` a factor with levels `Hombre` `Mujer`

`Familia` a numeric vector

`CasaPropia` a factor with levels `No` `Si`

`AnosEmpleo` a numeric vector

`Sueldo` a numeric vector

`Prestamo` a factor with levels `No` `Si`

`x31` a numeric vector

`x32` a numeric vector

`x33` a numeric vector

Source

Originated by Edgar Acuna

Examples

```
data(arboleje1)
library(rpart)
rpart(Prestamo~.,data=arboleje1,method="class")
```

autompg *The Auto MPG dataset*

Description

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition. Six instances containing missing values had been deleted.

Usage

```
data("autompg")
```

Format

A data frame with 392 observations on the following 8 variables.

`mpg` a numeric vector indicating the mileage per gallon comsumption

`cylinders` a numeric vector indicating the automobile's cylinders

`displacement` a numeric vector

`horsepower` a numeric vector

`weight` a numeric vector

`acceleration` a numeric vector

`modelyear` a numeric vector indicating the automobile's year model

`maker` a numeric vector. The value 1 is for american automobile, the value 2 is for european automobile, and the value 3 is for an asian automobile

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: data(autompg)
maxlof(autompg,name="maxlof")

## End(Not run)
```

baysout *Outlier detection using Bay and Schwabacher's algorithm.*

Description

This function implements the algorithm for outlier detection found in Bay and Schwabacher(2003). The algorithm assigns an outlyingness measure to each observation and returns the indexes of those observations having the largest measures. The number of outliers to be returned is specified by the user.

Usage

```
baysout(D, blocks = 10, nclass=0, k = 3, num.out = 10)
```

Arguments

D	the dataset under study
blocks	the number of sections in which to divide the entire dataset. It must be at least as large as the number of outliers requested.
nclass	To find the outliers without taking in cnsideration the feature class enter 0. To find the outliers for a given class enter the class number.
k	the number of neighbors to find for each observation
num.out	the number of outliers to return

Value

num.out	Returns a two column matrix containing the indexes of the observations with the top num.out outlyingness measures. A plot of the top candidates and their measures is also displayed.

Author(s)

Caroline Rodriguez(2004). Modified by Elio Lozano (2005) and Edgar Acuna (2015)

References

Bay, S.D., and Schwabacher (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule.

Examples

```
#---- Outliers detection using the Bay's algorithm----
data(bupa)
bupa.out=baysout(bupa[bupa[,7]==1,1:6],blocks=10,num.out=10)
```

`breastw` *The Breast Wisconsin dataset*

Description

This is the Breast Wisconsin dataset from the UCI Machine Learning Repository. This dataset has 699 instances, sixteen of them with missing values, 9 predictor attributes and one class attribute assuming values 1(benign tumor) and 2(malign tumor).

Usage

```
data(breastw)
```

Format

A data frame with 699 observations on the following 10 variables.

V1 Clump Thickness

V2 Uniformity of Cell Size

V3 Uniformity of Cell Shape

V4 Marginal Adhesion

V5 Single Epithelial Cell Size

V6 Bare Nuclei

V7 Bland Chromatin

V8 Normal Nucleoli

V9 Mitoses

V10 Class: 1 for benign, 2 for Malign

Details

All the features assume values in the range 1-10. The dataset contains 699 observations with 16 of them having missing values. It is recommended to impute these values been before further analysis.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
#Detecting outliers in class-1 using the LOF algorithms---
data(breastw)
ce.impute(breastw,"mean",1:9)
```

bupa *The Bupa dataset*

Description

The Bupa dataset

Usage

```
data(bupa)
```

Format

A data frame with 345 observations on the following 7 variables.

V1 mean corpuscular volume

V2 alkaline phosphotase

V3 alamine aminotransferase

V4 aspartate aminotransferase

V5 gamma-glutamyl transpeptidase

V6 number of half-pint equivalents of alcoholic beverages drunk per day

V7 The class variable (two classes)

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: #---Sequential forward feature selection using the lda classifier---
data(bupa)
sfs(bupa,"lda",repet=10)

## End(Not run)
```

ce.impute *Imputation in supervised classification*

Description

This function performs data imputation in datasets for supervised classification by using mean, median or knn imputation methods. The mode is used when the attribute is nominal

Usage

```
ce.impute(data, method = c("mean", "median", "knn"), atr,
 nomatr = rep(0, 0), k1 = 10)
```

Arguments

data	the name of the dataset
method	the name of the method to be used
atr	a vector identifying the attributes where imputations will be performed
nomatr	a vector identifying the nominal attributes
k1	the number of neighbors to be used for the knn imputation

Value

Returns a matrix without missing values.

Note

A description of all the imputations carried out may be stored in a report that is later saved to the current workspace. To produce the report, lines at the end of the code must be uncommented. The report objects name starts with Imput.rep.

Author(s)

Caroline Rodriguez

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

See Also

clean

Examples

```
data(hepatitis)
#--------Median Imputation-----------
#ce.impute(hepatitis,"median",1:19)
#--------knn Imputation--------------
hepa.imputed=ce.impute(hepatitis,"knn",k1=10)
```

`ce.mimp` *Mean or median imputation*

Description

A function that detects the location of missing values by class, then imputes the missing values that occur in the features, using mean or median imputation, as selected by the user. If the feature is nominal then imputation is done using the mode.

Usage

```
ce.mimp(w.cl, method = c("mean", "median"), atr, nomatr = 0)
```

Arguments

`w.cl`	dataset with missing values.
`method`	either "mean" or "median"
`atr`	list of relevant features
`nomatr`	list of nominal features, imputation is done using mode

Value

`w.cl`	the original matrix with values imputed

Note

A description of all the imputations carried out may be stored in a report that is later saved to the current workspace. To produce the report, lines at the end of the code must be uncommented. The report objects name starts with Imput.rep.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

Examples

```
data(hepatitis)
#--------Mean Imputation----------
hepa.mean.imp=ce.impute(hepatitis,"mean",1:19)
```

---

census *census*

Description

This is the census (also known as adult) dataset. It is used to predict the salary of a person based on socio-demographis and economic predictors.

Usage

```
data("census")
```

Format

A data frame with 32561 observations on the following 15 variables.

`age` a numeric vector

`employment` a factor with levels `Federal-gov` `Local-gov` `Never-worked` `Private` `Self-emp-inc` `Self-emp-not-inc` `State-gov` `Without-pay`

`a3` a numeric vector

`education` a factor with levels `10th` `11th` `12th` `1st-4th` `5th-6th` `7th-8th` `9th` `Assoc-acdm` `Assoc-voc` `Bachelors` `Doctorate` `HS-grad` `Masters` `Preschool` `Prof-school` `Some-college`

`education.num` a numeric vector

`marital.status` a factor with levels `Divorced` `Married-AF-spouse` `Married-civ-spouse` `Married-spouse-absent` `Never-married` `Separated` `Widowed`

`job` a factor with levels `Adm-clerical` `Armed-Forces` `Craft-repair` `Exec-managerial` `Farming-fishing` `Handlers-cleaners` `Machine-op-inspct` `Other-service` `Priv-house-serv` `Prof-specialty` `Protective-serv` `Sales` `Tech-support` `Transport-moving`

`relationship` a factor with levels `Husband` `Not-in-family` `Other-relative` `Own-child` `Unmarried` `Wife`

`race` a factor with levels `Amer-Indian-Eskimo` `Asian-Pac-Islander` `Black` `Other` `White`

`gender` a factor with levels `Female` `Male`

`a11` a numeric vector

`a12` a numeric vector

`hours.per.week` a numeric vector

`native.country` a factor with levels `Cambodia Canada China Columbia Cuba Dominican-Republic Ecuador El-Salvador England France Germany Greece Guatemala Haiti Holand-Netherlands Honduras Hong Hungary India Iran Ireland Italy Jamaica Japan Laos Mexico Nicaragua Outlying-US(Guam-USVI-etc) Peru Philippines Poland Portugal Puerto-Rico Scotland South Taiwan Thailand Trinadad&Tobago United-States Vietnam Yugoslavia`

`salary` a factor with levels `<=50K >50K`

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: data(census)
imagmiss(census,"census")

## End(Not run)
```

---

chiMerge *Discretization using the Chi-Merge method*

Description

This function performs supervised discretization using the Chi Merge method.

Usage

```
chiMerge(data, varcon, alpha = 0.1,out=c("symb","num"))
```

Arguments

`data`	The name of the dataset to be discretized
`varcon`	Vector of continuous variables
`alpha`	The significance level
`out`	To get the discretized data in numerical format enter "num". To get the discretized data in interval format enter "symb".

Details

In case of datasets containing negative values apply first a range normalization to change the range of the attributes values to an interval containing positive values. The discretization process becomes slow when the number of variables increases (say for more than 100 variables).

Value

discdata	A new data matrix containing the discretized features

Author(s)

Edgar Acuna, Jaime Porras, and Carlos Lopez

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

disc.ef, disc.ew,disc.1r,disc.mentr

Examples

```
#-----Discretization using the ChiMerge method
data(iris)
iris.disc=chiMerge(iris,1:4,alpha=0.05,out="num")
#-----Applying chiMerge a dataset containing negative values
#data(ionosphere)
#normionos=rangenorm(ionosphere,"mmnorm")
#ionos.disc=chiMerge(normionos,1:32)
```

---

circledraw *circledraw*

Description

This function draws a circle using the polygon function from the graphics package. It is an auxiliary function used by radviz2d.

Usage

```
circledraw (numpts = 200, radius = 1)
```

Arguments

numpts	Number of edges of the polygon, default is 200.
radius	Radius of the circle to be drawn, default is 1.

Details

A circle of a specified radius is drawn by the polygon function of the graphics library by constructing a polygon with numpts number of edges. It is intended to be an auxiliary function for the radviz2d visualization.

Value

Displays a circle of radius = radius.

Author(s)

Caroline Rodriguez

Examples

```
#----Circledraw examples
circledraw()
```

`clean` *Dataset's cleaning*

Description

A function to eliminate rows and columns that have a percentage of missing values greater than the allowed tolerance.

Usage

```
clean(w, tol.col = 0.5, tol.row = 0.3, name = "")
```

Arguments

`w`	the dataset to be examined and cleaned
`tol.col`	maximum ratio of missing values allowed in columns. The default value is 0.5. Columns with a larger ratio of missing will be eliminated unless they are known to be relevant attributes.
`tol.row`	maximum ratio of missing values allowed in rows. The default value is 0.3. Rows with a ratio of missing that is larger that the established tolerance will be eliminated.
`name`	name of the dataset to be used for the optional report

Details

This function can create an optional report on the cleaning process if the comment symbols are removed from the last lines of code. The report is returned to the workspace, where it can be reexamined as needed. The report object's name begins with: Clean.rep.

Value

`w`	the original dataset, with missing values that were in relevant variables imputed

Author(s)

Caroline Rodriguez

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

See Also

ce.impute

Examples

```
#-----Dataset cleaning-----
data(hepatitis)
hepa.cl=clean(hepatitis,0.5,0.3,name="hepatitis-clean")
```

---

colon *Alon et al.'s colon dataset*

Description

This is Alon et al.'s Colon cancer dataset which contains information on 62 samples for 2000 genes. The samples belong to tumor and normal colon tissues.

Usage

```
data(colon)
```

Format

A data frame with 62 observations for 2000 genes. An additional column contains the tissue classes.

Source

The data is available at:

- http://microarray.princeton.edu/oncology/

References

Alon U, Barkai N, Notterman DA, Gish, K, Ybarra, S. Mack, D and Levine, AJ. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA. 96. p. 6745-6750.

Examples

```
#Detecting the top 5 outliers in class-2 using the LOF algorithm
data(colon)
colon2.lof=maxlof(colon[colon[,2001]==2,],"colon-class2")
colon2.lof[order(colon2.lof,decreasing=TRUE)][1:5]
```

`combinations` *Constructing distinct permutations*

Description

A function for constructing the minimal set of permutations of the elements in the sequence 1:numcol as described by Wegman in Hyperdimensional Data Analysis(1999)

Usage

```
combinations(numcol)
```

Arguments

`numcol`	A value representing the number of columns in a matrix

Value

A matrix in which each column represents a distinct permutation of the sequence 1:numcol

Author(s)

Caroline Rodriguez

References

Wegman, E. (1990), Hyperdimensional data analysis using parallel coordinates, Journal of the American Statistical Association, 85, 664-675.

crossval *Cross validation estimation of the misclassification error*

Description

Computation of the misclassification error for the LDA, KNN and rpart classifiers by cross validation

Usage

```
crossval(data, nparts = 10, method = c("lda", "knn", "rpart","logistic","naiveBayes"),
kvec = 5,maxwts=2500, repet)
```

Arguments

data	The name of the dataset
nparts	The number of folds in which the dataset is divided. By default nparts=10.
method	The name of the classifier to be used: LDA,KNN, Rpart.
kvec	The number of nearest neighbors to be used for the KNN classifier.
maxwts	The maximum number of iteratios to be used in th computation of the logistic regression
repet	The number of repetitions

Value

Returns the mean misclassification crossvalidation error of the classifier obtained on a given number of repetitions

Author(s)

Edgar Acuna

See Also

cv10log, cv10mlp

Examples

```
#------10-fold crossvalidation error using the LDA classifier---
data(bupa)
crossval(bupa,method="lda",repet=10)
## Not run: #------5-fold crossvalidation error using the knn classifier---
data(colon)
crossval(colon,nparts=5,method="knn",kvec=3,repet=5)

## End(Not run)
```

crx *crx*

Description

The Australian credit Aproval dataset form the Statlog Project

Usage

```
data("crx")
```

Format

A data frame with 690 observations on the following 16 variables.

`V1` a factor with levels `a b`

`V2` a numeric vector

`V3` a numeric vector

`V4` a factor with levels `l u y`

`V5` a factor with levels `g gg p`

`V6` a factor with levels `aa c cc d e ff i j k m q r w x`

`V7` a factor with levels `bb dd ff h j n o v z`

`V8` a numeric vector

`V9` a factor with levels `f t`

`V10` a factor with levels `f t`

`V11` a numeric vector

`V12` a factor with levels `f t`

`V13` a factor with levels `g p s`

`V14` a numeric vector

`V15` a numeric vector

`V16` a numeric vector

Details

This dataset contains missing values.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
data(crx)
ec.knnimp(crx,nomatr=c(1,4:7,9,10,12,13),k=3)
```

cv10knn2 *Auxiliary function for sequential feature selection*

Description

This function finds the number of instances correctly classified by the knn classifier, using 10-fold cross validation, with one repetition

Usage

```
cv10knn2(data, kvec)
```

Arguments

data	The name of the dataset
kvec	The number of neighbors

Author(s)

Edgar Acuna

See Also

crossval

cv10lda2 *Auxiliary function for sequential forward selection*

Description

This function finds the number of instances correctly classified by the Linear Discriminant classifier using 10 fold cross validation with one repetition.

Usage

```
cv10lda2(data)
```

Arguments

data	The name of the dataset

Author(s)

Edgar Acuna

See Also

crossval

cv10log *10-fold cross validation estimation error for the classifier based on logistic regression*

Description

10-fold cross validation estimation of the misclassification error for the classifier based on logistic regression

Usage

```
cv10log(data, repet,maxwts=2500)
```

Arguments

data	The name of the dataset
repet	The number of repetitions
maxwts	The maximum number of weights to be estimated. It must be an integer greater than the number of predictors of the dataset.

Value

The mean cross validation error for the classifier based on logistic regression using the number of repetitions

Author(s)

Edgar Acuna

References

Ripley, B.D. (1996). Pattern recognition and Neural networks. Cambridge University Press

Venables,W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also

crossval, cv10mlp

Examples

```
#-----cross validation error for the logistic classifier-------
data(bupa)
cv10log(bupa,5)
```

`cv10mlp` *10-fold cross validation error estimation for the multilayer perceptron classifier*

Description

10-fold cross validation estimation error for the multilayer perceptron classifier.

Usage

```
cv10mlp(data, units, decay = 0, maxwts = 1000, maxit = 100,
repet)
```

Arguments

`data`	The name of the dataset
`units`	The number of units in the hidden layer
`decay`	The decay parameter
`maxwts`	The maximum number of weights to be estimated in the network
`maxit`	The maximum number of iterations
`repet`	The number of repetitions

Value

Returns the mean cross validation for the multilayer perceptron classifier.

Author(s)

Edgar Acuna

References

Ripley, B.D. (1996). Pattern recognition and Neural networks. Cambridge University Press

Venables,W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also

`crossval`, `cv10log`

Examples

```
## Not run: #-----cross validation using the MLP classifier---
data(heartc)
heartc=ce.impute(heartc,"mean",1:13)
cv10mlp(heartc,25,decay=0.1,maxwts=1000,maxit=100,repet=2)

## End(Not run)
```

cv10rpart2 *Auxiliary function for sequential feature selection*

Description

This function finds the number of instances correctly classified by the decision tree classifier, rpart, using 10-fold cross validation and one repetition.

Usage

```
cv10rpart2(datos)
```

Arguments

datos	The name of the dataset

Author(s)

Edgar Acuna

See Also

crossval

cvnaiveBayesd *Crossvalidation estimation error for the naive Bayes classifier.*

Description

This function computes the crossvalidaton error for a naive bayes classifier after discretization

Usage

```
cvnaiveBayesd(data, repet, method = c("ew", "ef", "1R", "chiMerge"))
```

Arguments

data	The dataset
repet	The number of repetions.
method	The discretezation method to be used, bins with equal widths (ew), bins with equal frequency (ef), Holte's oneR (1R) and chiMerge.

Details

Uses 10-fold crossvalidation.

Value

Returns the mean misclassification crossvalidation error of the classifier obtained on a given number of repetions

Author(s)

Edgar Acuna

See Also

crossval

Examples

```
data(diabetes)
library(e1071)
cvnaiveBayesd(diabetes,3,method="ew")
```

---

decscale *Decimal Scaling*

Description

This is a function to apply decimal scaling to a matrix or dataframe. Decimal scaling transforms the data into [-1,1] by finding k such that the absolute value of the maximum value of each attribute divided by 10\^k is less than or equal to 1.

Usage

```
decscale(data)
```

Arguments

data	The dataset to be scaled

Details

Uses the scale function found in the R base package.

Value

decdata	The original matrix that has been scaled by decimal scaling

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
data(sonar)
def=par(mfrow=c(2,1))
plot(sonar[,2])
dssonar=decscale(sonar)
plot(dssonar[,2])
par(def)
```

`diabetes` *The Pima Indian Diabetes dataset*

Description

This is the Pima Indian diabetes dataset from the UCI Machine Learning Repository.

Usage

```
data(diabetes)
```

Format

A data frame with 768 observations on the following 9 variables.

V1 Number of times pregnant

V2 Plasma glucose concentration (glucose tolerance test)

V3 Diastolic blood pressure (mm Hg)

V4 Triceps skin fold thickness (mm)

V5 2-Hour serum insulin (mu U/ml)

V6 Body mass index (weight in kg/(height in m)\^2)

V7 Diabetes pedigree function

V8 Age (years)

V9 Class variable (1:tested positive for diabetes, 0: tested negative fro diabetes)

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
#---Feature selection using SFS with the LDA classifier--
data(diabetes)
sfs(diabetes,"lda",repet=1)
```

disc.1r *Discretization using the Holte's 1R method*

Description

This function performs supervised discretization using the Holte's 1R method

Usage

```
disc.1r(data, convar, binsize = 15, out=c("symb","num"))
```

Arguments

data	The name of the dataset to be discretized
convar	A vector containing the continuous features
binsize	The number of instances per bin
out	To get the discretized dataset in a numerical format write "num". To get the discretized in an interval format write "symb"

Value

Returns a new data matrix with discretized values

Author(s)

Edgar Acuna

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

disc.ew,disc.ef, chiMerge, disc.mentr

Examples

```
#-----Discretization using the Holte's 1r method
data(bupa)
disc.1r(bupa,1:6,out="symb")
```

disc.ef *Discretization using the method of equal frequencies*

Description

Unsupervised discretization using intervals of equal frequencies

Usage

```
disc.ef(data, varcon, k,out=c("symb","num"))
```

Arguments

data	The dataset to be discretized
varcon	A vector containing the continuous features
k	The number of intervals to be used
out	To get the discretized dataset in a numerical format write "num". To get the discretized in an interval format write "symb"

Value

Returns a new data matrix with discretized values.

Author(s)

Edgar Acuna

References

Kantardzic M. (2003). Data Mining: Concepts, Models, methods, and Algorithms. John Wiley. New York.

See Also

disc.1r, disc.ew,chiMerge

Examples

```
#Discretization using the equal frequency method
data(bupa)
bupa.disc.ef=disc.ef(bupa,1:6,8,out="symb")
```

disc.ew *Discretization using the equal width method*

Description

Unsupervised discretization using intervals of equal width. The widths are computed using Scott's formula.

Usage

```
disc.ew(data, varcon,out=c("symb","num"))
```

Arguments

data	The name of the dataset containing the attributes to be discretized
varcon	A vector containing the indexes of the attributes to be discretized
out	To get the discretized dataset in a numerical format write "num". To get the discretized in an interval format write "symb"

Value

Returns a new data matrix with discretized values.

Author(s)

Edgar Acuna

References

Venables, W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth edition, Springer

See Also

disc.ef, disc.1r,chiMerge,disc.mentr

Examples

```
#----Discretization using the equal frequency method
data(bupa)
bupa.disc.ew=disc.ew(bupa,1:6,out="num")
```

disc.mentr *Discretization using the minimum entropy criterion*

Description

This function discretizes the continuous attributes of a data frame using the minumum entropy criterion along with the minimum description length as stopping rule.

Usage

```
disc.mentr(data, varcon,out=c("symb","num"))
```

Arguments

data	The name of the dataset to be discretized
varcon	A vector containing the indices of the columms to be discretized
out	To get the data discretized in numerical form enter "num". To get the data discretized in interval form enter "symb"

Value

Returns a matrix containing only discretized features.

Author(s)

Luis Daza(2006) and Edgar Acuna(2015)

References

Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. ML-95.

See Also

disc.1r, disc.ew,disc.ef,chiMerge

Examples

```
## Not run:
#----Discretization using the entropy with Minimum Description Length.
data(bupa)
bupa.disc=disc.mentr(bupa,1:6,out="num")

## End(Not run)
```

disc2 *Auxiliary function for performing discretization using equal frequency*

Description

This function is called by the disc.ef function in the dprep library.

Usage

```
disc2(x, k, out=c("symb","num"))
```

Arguments

x	A numerical vector
k	The number of intervals
out	To get the discretized data in numerical fromat enter "num". To get the discretized data in interval format enter "symb".

Author(s)

Edgar Acuna

See Also

disc.ef

discretevar *Performs Minimum Entropy discretization for a given attribute*

Description

This function carries out ME discretization for a given attribute of a dataset. It is also called from within the function discr.mentr.

Usage

```
discretevar(data, var, n, p)
```

Arguments

data	The name of the dataset
var	The column where the attribute to be discretized is located
n	The number of rows of the dataset
p	The number of columns of the dataset

Author(s)

Luis Daza

See Also

disc.mentr

---

dist.to.knn *Auxiliary function for the LOF algorithm.*

Description

This function returns an object in which columns contain the indices of the first k neighbors followed by the distances to each of these neighbors.

Usage

```
dist.to.knn(dataset, neighbors)
```

Arguments

dataset	The name of the dataset
neighbors	The number of neighbors

Author(s)

Caroline Rodriguez

See Also

maxlof

---

distancia *Vector-Vector Euclidiean Distance Function*

Description

Finds the euclidean distance between two vectors x and y, or the vector x and the matrix y

Usage

```
distancia(x, y)
```

Arguments

x	A numeric vector
y	A numeric vector or matrix

Details

Does not support missing values.

Value

distancia	The result is a numeric value representing the Euclidean distance between x and y, or a row matrix representing the Euclidean distance between x and each row of y.

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Calculating distances
x=rnorm(4)
y=matrix(rnorm(12),4,3)
distancia(x,y[,1])
distancia(x,y)
```

---

distancia1 *Vector-Vector Manhattan Distance Function*

Description

Finds the Manhattan distance between two vectors x and y, or the vector x and the matrix y

Usage

```
distancia1(x, y)
```

Arguments

x	A numeric vector
y	A numeric vector or matrix

Details

Does not support missing values.

Value

distancia	The result is a numeric value representing the Manhattan distance between x and y, or a row matrix representing the Euclidean distance between x and each row of y.

Author(s)

Edgar Acuna

Examples

```
#---- Calculating distances
x=rnorm(4)
y=matrix(rnorm(12),4,3)
distancia1(x,y[,1])
distancia1(x,y)
```

---

ec.knnimp *Imputation using k-nearest neighbors.*

Description

This function imputes missing values by knn imputation.

Usage

```
ec.knnimp(data, nomatr=0, k=10)
```

Arguments

data	matrix containing relevant variables and classes
nomatr	list of nominal attributes
k	number of neighbors to use for imputation

Value

r	matrix with missing values imputed

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
data(hepatitis)
hepa.knnimp=ec.knnimp(hepatitis,nomatr=c(1,3:14),k=10)
```

eje1dis *Basic example for discriminant analysis*

Description

This data frame contains information about 32 students. The first two columns contain their grades obtained on the first two exams and the last column of the dataset contains the classes: P=Pass, and F=Fail

Usage

```
data(eje1dis)
```

Format

A data frame with 32 observations on the following 3 variables.

E1 Grade on the first exam

E2 Grade on the second exam

Class The class vector:P=Pass, F=Fail

Source

Data obtained from Edgar Acuna:

- http://academic.uprm.edu/eacuna/datosclass.html

Examples

```
#---- Performing 10-fold cross validation using the LDA classifier-----
data(eje1dis)
crossval(eje1dis,10,"lda",repet=5)
```

finco *FINCO Feature Selection Algorithm*

Description

This function selects features using the FINCO algorithm. The dataset must contain only discretized values.

Usage

```
finco(data,level)
```

Arguments

data	Name of the dataset containing the discretized values
level	Minimum inconsistency level

Details

The level value must be greater than the inconsistency of the whole dataset, which first must be discretized. The function inconsist included in this library computes inconsistencies. A small value for level yields a greater number of selected features.

Value

varselec	Index of selected features
inconsis	Inconsistency rates of the selected features

Author(s)

Edgar Acuna

References

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Acuna, E., Coaquira, F. and Gonzalez, M. (2003). A comparison of feature selection procedures for classifiers based on kernel density estimation. Proc. of the Int. Conf. on Computer, Communication and Control technologies, CCCT03. VolI. p. 468-472. Orlando, Florida.

See Also

inconsist,lvf

Examples

```
#---- Feature Selection with FINCO
data(iris)
iris.discew=disc.ew(iris,1:6,out="num")
inconsist(iris.discew)
finco(iris.discew,0.05)
```

heartc *The Heart Cleveland dataset*

Description

This dataset contains information concerning heart disease diagnosis. The data was collected from the Cleveland Clinic Foundation, and it is available at the UCI machine learning Repository. Six instances containing missing values.

Usage

```
data(heartc)
```

Format

A data frame with 297 observations on the following 14 variables.

V1 age(continuous)

V2 sex

V3 cp, chest pain type:1,2,3,4

V4 trestbps: resting blood pressure(continuous)

V5 cholesterol(continuous)

V6 fps: fatsing blood sugar>120? yes=1, no =0

V7 restecg: resting electrocardiographic results, 0,1, 2

V8 thalach: maximum heart rate achieved(continuous)

V9 exang: exercise induced angina (1 = yes; 0 = no)

V10 oldpeak = ST depression induced by exercise relative to rest (continuous)

V11 slope: the slope of the peak exercise ST segment

V12 ca: number of major vessels (0-3) colored by flourosopy

V13 thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

V14 diagnosis of heart disease: 1: < 50 2: > 50

Details

This dataset contains six instances having missing values. It is recommended to impute these values before applying other tasks. This dataset includes continuous, binomial, nominal, and ordinal features.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
#----Detecting outliers using the Relief---
data(heartc)
imagmiss(heartc,"heart-Cleveland")
```

`hepatitis` *The hepatitis dataset*

Description

This is the hepatitis dataset from the UCI. The data was donated by Gail Gong.

Usage

```
data(hepatitis)
```

Format

A data frame with 155 observations on the following 20 variables. This dataset contains a large number of missing values.

V1 Histology:no,yes

V2 age

V3 sex: male,female

V4 steroid:no,yes

V5 antivirals:no,yes

V6 fatigue:no, yes

V7 malaise:no, yes

V8 anorexia:no, yes

V9 liver big:no,yes

V10 liver firm:no,yes

V11 spleen palpable: no, yes

V12 spiders:no,yes

V13 ascites:no,yes

V14 Varices:no,yes

V15 Bilirubin

V16 alk phosphate

V17 sgot

V18 Albumin

V19 Protime

V20 Class:Die, Live

Details

The original dataset has the class labels in the first column.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

References

Diaconis,P. & Efron,B. (1983). Computer-Intensive Methods in Statistics. Scientific American, Volume 248.

Examples

```
#------Report and plot of missing values ------
data(hepatitis)
imagmiss(hepatitis,"Hepatitis")
```

---

imagmiss *Visualization of Missing Data*

Description

Function to create a graph of the observations of the dataset leaving white gaps where data is missing.

Usage

```
imagmiss(data, name = "")
```

Arguments

data	The dataset containing missing values
name	The name of dataset to be used in title of plot

Details

The main idea is to use the original dataset to create a temporary dataset containing 1 if a value is found or 0 if the value is missing. The temporary data set is graphed by column, changing color for each feature and leaving a blank horizontal line if a value is missing. Assumes classes are in the last column, and removes the column containing the classes before plotting. A report that describes the percentage of missing values in the data set is provided once the visualization is complete.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648.

Examples

```
#---- Plotting datasets with missing values
data(hepatitis)
imagmiss(hepatitis, "hepatitis")
```

---

inconsist *Computing the inconsistency measure*

Description

This function computes the inconsistency of a discretized dataset.

Usage

```
inconsist(data)
```

Arguments

data	a discretized dataset

Details

This function requires the function row.matches included in this environment package, and the function unique from the base library.

Value

incon	the inconsistency measure of the dataset

Author(s)

Edgar Acuna

References

Dash M., Liu H, and Motoda, H. (1998). Consistency Based Feature Selection Pacific-Asia Conference on Knowledge Discovery and Data Mining

See Also

finco, lvf

Examples

```
##---- Calculating Inconsistency ----
data(bupa)
bupa.discew=disc.ew(bupa,1:6)
inconsist(bupa.discew)
```

---

ionosphere *The Ionosphere dataset*

Description

The Ionosphere dataset from the UCI Machine Learning Repository

Usage

```
data(ionosphere)
```

Format

A data frame with 351 observations on the following 33 variables.

Details

The original dataset contains 34 predictors, but we have eliminated the two first features, because the first feature had the same value in one of the classes and the second feature assumes the value 0 in all observations.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
#---Outlier detection in  ionosphere class-1 using the Mahalanobis distiance----
data(ionosphere)
mahaout(ionosphere,1)
```

knneigh.vect *Auxiliary function for computing the LOF measure.*

Description

Function that returns the distance from a vector "x" to its k-nearest-neighbors in the matrix "data"

Usage

```
knneigh.vect(x, data, k)
```

Arguments

x	A given instance of the data matrix
data	The name of the data matrix
k	The number of neigbors

Author(s)

Caroline Rodriguez

See Also

maxlof

knngow *K-nn classification using Gower distance*

Description

This function performs classification using the k- nearest neighbors but using Gower distance rather than Euclidean distance. It is recommended if the dataset has niminal and continuous attributes.

Usage

```
knngow(train, test, k)
```

Arguments

train	A matrix containing the training dataset.
test	A matrix containing the test dataset.
k	The number of neighbors to be used.

Value

predclass A vector of predited classes

Author(s)

Edgar Acuna

See Also

acugow

Examples

```
## Not run: data(crx)
knngow(crx,crx,3)
## End(Not run)
```

---

landsat *The landsat Satellite dataset*

Description

This is the Landsat Satellite dataset from the Stalog project. The training and test dataset have been joined to form a single dataset

Usage

```
data("landsat")
```

Format

A data frame with 6435 observations and 37 variables.

Details

The training set has 4435 intances y el test set kas 2000 instances.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: data(landsat)
relief(landsat,200,0,3)

## End(Not run)
```

`lofactor` *Local Outlier Factor*

Description

A function that finds the local outlier factor (Breunig et al.,2000) of the matrix "data" using k neighbors. The local outlier factor (LOF) is a measure of outlyingness that is calculated for each observation. The user decides whether or not an observation will be considered an outlier based on this measure. The LOF takes into consideration the density of the neighborhood around the observation to determine its outlyingness.

Usage

```
lofactor(data, k)
```

Arguments

`data`	The data set to be explored
`k`	The kth-distance to be used to calculate the LOF's.

Details

The LOFs are calculated over a range of values, and the max local outlier factor is determined over this range.

Value

`lof`	A vector with the local outlier factor of each observation

Author(s)

Caroline Rodriguez

References

Breuning, M., Kriegel, H., Ng, R.T, and Sander. J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

Examples

```
#---- Detecting the top 10 outliers using the LOF algorithm----
data(bupa)
bupa.lof=lofactor(bupa,10)
```

lvf *Las Vegas Filter*

Description

Las Vegas Filter uses a random generation of subsets and an inconsistency measure as the evaluation function to determine the relevance of features in the dataset.

Usage

```
lvf(data, lambda, maxiter)
```

Arguments

data	Name of the discretized dataset
lambda	Threshold for the inconsistency
maxiter	Maximum number of iterations

Details

If the dataset has continuous variables, these must first be discretized. This package includes four discretization methods. A value of lambda close to the inconsistency of the whole dataset yields a large number of selected features, a large lambda yields few selected features.

Value

bestsubset	The best subset of features

Author(s)

Edgar Acuna

References

LIU, H. and SETIONO, R. (1996). A probabilistic approach to feature selection: a filter solution. Proc. of the thirteenth International Conference of Machine Learning, 319-337.

See Also

disc.ew,inconsist,finco

Examples

```
#---- LVF method ----
data(iris)
iris.discew=disc.ew(iris,1:4,out="num")
inconsist(iris.discew)
lvf(iris.discew,0,100)
```

mahaout *Multivariate outlier detection through the boxplot of the Mahalanobis distance*

Description

This function finds multivariate outliers by constructing a boxplot of the Mahalanobis distance of all the instances.

Usage

```
mahaout(data, nclass=0, plot = TRUE)
```

Arguments

data	Name of the dataset
nclass	Number of the class to check for outliers. By default nclass=0 meaning the column of classes it is not used.
plot	Logical value. If plot=T a plot of the mahalanobis distance is drawn

Details

uses cov.rob function from the MASS library

Value

Returns a list of top outliers according to their Mahalanobis distance and a list of all the instances ordered according to their Mahalanobis distance.

If Plot=T, a plot of the instances ranked by their Mahalanobis distance is provided.

Author(s)

Edgar Acuna

References

Rousseeuw, P, and Leroy, A. (1987). Robust Regression and outlier detection. John Wiley & Sons. New York.

See Also

robout

Examples

```
#---- Detecting outliers using the Mahalanobis distance----
data(bupa)
mahaout(bupa,1)
```

mardia *The Mardia's test of normality*

Description

Performs the Mardia's test to check for multivariate normality

Usage

```
mardia(data)
```

Arguments

data	The dataset containing the features for which multivariate normality is going to be tested. The last column contains the class. In case of unsupervised data add a dummy columnn of ones. In case of regression data, transform the response column in a column of ones

Value

Returns the p-values for the corresponding third and fourth moments of the multivariate normal distribution.

Author(s)

Edgar Acuna

References

Mardia, K.V. (1985). "Mardia's Test of Multinormality," in S. Kotz and N.L. Johnson, eds., Encyclopedia of Statistical Sciences, vol. 5 (NY: Wiley), pp. 217-221.

See Also

vvalen

Examples

```
#-----Mardia test for supervised data-----
data(iris)
mardia(iris)
```

maxlof *Detection of multivariate outliers using the LOF algorithm*

Description

A function that detects multivariate outliers using the local outlier factor for a matrix over a range of neighbors called minpts.

Usage

```
maxlof(data, name = "", minptsl = 10, minptsu = 20)
```

Arguments

data	Dataset for outlier detection
name	Name of dataset used in the graph title.
minptsl	Lower bound for the number of neighbors
minptsu	Upper bound for the number of neighbors

Details

Calls on the function "lofactor" to compute the local outlier factor for each integer number of neighbors in the range [minptsl, minptsu]. Also displays a plot of the factors for each observation of the dataset. In the plot, the user should seek to identify observations with large gaps between outlyingness measures. These would be candidates for outliers.

Value

maxlofactor	A vector containing the index of each observation of the dataset and the corresponding local outlier factor.

Author(s)

Caroline Rodriguez

References

Breuning, M., Kriegel, H., Ng, R.T, and Sander. J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

Examples

```
## Not run: #Detecting top 10 outliers in class number 1 of Breastw using the LOF algorithm
data(breastw)
breastw=ce.impute(breastw,"median",1:9)
breastw1.lof=maxlof(breastw[breastw[,10]==1,],name="Breast-Wisconsin",30,40)
breastw1.lof[order(breastw1.lof,decreasing=TRUE)][1:10]

## End(Not run)
```

midpoints1 *Auxiliary function for computing minimun entropy discretization*

Description

This function finds out a vector of midpoints of a given vector. In case of a type yield zero. Further adds zero as lasta entry of the midpoints vector.

Usage

```
midpoints1(x)
```

Arguments

x	A numerical vector

Author(s)

Luis Daza and Edgar Acuna

See Also

disc.mentr

mmnorm *Min-max normalization*

Description

This is a function to apply min-max normalization to a matrix or dataframe.

Usage

```
mmnorm(data,minval=0,maxval=1)
```

Arguments

data	The dataset to be normalized, including classes
minval	The minimun value of the transformed range
maxval	The maximum value of the transformed range

Details

Min-max normalization subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. These new values are multiplied by the new range of the attribute and finally added to the new minimum value of the attribute. These operations transform the data into a new range, generally [0,1]. The function removes classes (assuming they are in last column) before normalization, and returns a normalized data set, complete with classes. Uses the function scale from the base package.

Value

zdata3	The normalized dataset

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#---- Min-Max Normalization----
data(ionosphere)
ionos.minmax=mmnorm(ionosphere)
op=par(mfrow=c(2,1))
plot(ionosphere[,1])
plot(ionos.minmax[,1])
par(op)
```

---

mo3 *The third moment of a multivariate distribution*

Description

This function computes the third moment of a multivariate normal distribution. This result is used later on the Mardia's test for multivariate normality

Usage

```
mo3(data)
```

Arguments

data	The dataset containing the features of the multivariate vector for which the third moment will be computed. Do not include the class attribute for supervised datasets

Value

mo3 The third moment of the multivariate distribution

Author(s)

Edgar Acuna

See Also

mo4, mardia

Examples

```
## Not run: data(iris)
mo3(iris[,-5])

## End(Not run)
```

---

mo4 *The fourth moment of a multivariate distribution*

Description

This function computes the fourth moment of a multivariate distribution. This result is used later in the mardia's test for multivariate normality.

Usage

```
mo4(data)
```

Arguments

data The dataset containing the features of the multivariate vector for which the fourth moment will be computed. Do not include the class attribute for supervised datasets

Value

Returns the fourth moment.

Author(s)

Edgar Acuna

See Also

mo3, mardia

Examples

```
data(iris)
mo4(iris[,-5])
```

---

moda *Calculating the Mode*

Description

This function calculates the mode of a vector.

Usage

```
moda(x, na.rm = TRUE)
```

Arguments

x	A numeric vector
na.rm	A Boolean value that indicates the presence of missing values.

Details

The function returns the mode or modes of a vector. If a tie exists, all values that are tied are returned.

Value

moda	A numeric value representing the mode of the vector

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Calculating the mode ----
x=c(1,4,2,3,4,6,3,7,8,5,4,3)
moda(x)
```

near1 *Auxiliary function for the reliefcont function*

Description

This function finds the instance in the data matrix that is closest to a given instance x using Euclidean distance. It is assumed that all the attributes are continuous.

Usage

```
near1(x, data)
```

Arguments

x	A given instance
data	The name of the dataset

Author(s)

Edgar Acuna

See Also

near3,distancia

near3 *Auxiliary function for the reliefcat function*

Description

This function finds the instance in the data matrix that is closest to a given instance x sing teh Manhattan distance. The attributes can be either continuous or nominal.

Usage

```
near3(x, data)
```

Arguments

x	A given instance
data	The name of the dataset

Author(s)

Edgar Acuna

See Also

relief,distancia1

---

nnmiss *Auxiliary function for knn imputation*

Description

This function is required to perform k-nn imputation.

Usage

```
nnmiss(x, xmiss, ismiss, xnom, K = 1)
```

Arguments

x	A submatrix of complete rows from original matrix
xmiss	A row with a missing value
ismiss	A vector that indicates whether a value in xmiss is missing or not
xnom	A vector with indexes of nominal variables
K	The number of neighbors to use

Author(s)

Edgar Acuna

See Also

ce.impute

---

outbox *Detecting outliers through boxplots of the features.*

Description

This function detects univariate outliers simultaneously using boxplots of the features.

Usage

```
outbox(data, nclass)
```

Arguments

data	The dataset to be explored for outlier detection.
nclass	A value representing the class that will be explored.

Details

The function also displays a plot containing a boxplot for of the variables.

Value

out1	A list of the indices of the observations that are outside the extremes of the boxplot. The indices are given in a table format representing the number of columns in which the observation was identified as an outlier.

Author(s)

Edgar Acuna

Examples

```
#---- Identifying outliers in diabetes-class1 with boxplots----
data(diabetes)
outbox(diabetes,nclass=1)
```

---

parallelplot *Parallel Coordinate Plot*

Description

Constructs a parallel coordinate plot for a data set with classes in last column.

Usage

```
parallelplot(x, name = "", comb = -1, class = 0, obs = rep(0, 0), col = 2, lty = 1, ...)
```

Arguments

x	A matrix of numerical values with classes in last column
name	The name of data set as will appear in the graph title
comb	An integer that represents the number of one of the possible combinations for the columns of this matrix.
class	A value representing the class number to which the plot should be limited
obs	A list of one or more row numbers that are to be highlighted in the plot
col	A value that provides a choice of color for the plot (if plotting only one class)
lty	A value that provides a choice of line width for the plot (if plotting only one class)
...	Additional arguments for the matplot function

Details

This plot is not recommended for a large number of features (say more than 50). If comb=0, all distinct combinations of columns are graphed. If comb=-1 (default), the attributes are plotted in their original order, else comb should be equal to an integer that represents the number of one of the possible combinations for the columns of this matrix.

Value

A parallel coordinate plot of the data is produced.

Author(s)

Caroline Rodriguez

References

Wegman, E. (1990), Hyperdimensional data analysis using parallel coordinates, Journal of the American Statistical Association,85,664-675

See Also

starcoord, surveyplot

Examples

```
#---Parallel Coordinate Plot----
data(bupa)
parallelplot(bupa,"Bupa Dataset")
parallelplot(bupa,"Bupa Dataset",comb=0)
#parallelplot(bupa,"Bupa Dataset",comb=1,c(1,22,50))
```

---

radviz2d *Radial Coordinate Visualization*

Description

Radviz is a radial spring-based visualization that permits the visualization of n-dimensional datasets. Data attributes are equidistantly distributed along the circumference of a circle. Each data item is virtually connected to a spring that starts at the circle perimeter and ends on the data item. Each spring pulls the item with a force proportional to the item attribute value. Depending on the value of each attribute, the forces of the springs project each data item to a position inside the circle where the sum of the spring forces is equal to zero.

Usage

```
radviz2d(dataset, name = "")
```

Arguments

dataset	The dataset to be visualized.
name	The name of the dataset to be used in the graph title.

Details

Some features of this visualization are: 1) Points where all dimensional values have approximately the same value will lie close to the center. 2) If dimensional points lie opposite each other on the circle and have similar values than points will lie near the center. 3) If 1 or 2 dimensional values are greater, points will lie closer to those dimensional points. 4) Where a point will lie depends on the layout of the particular dimensions around the circle. 5) This is a non-linear projection from N-dimensions down to 2 dimensions 6) Certain symmetries of the data will be preserved.

The function assumes the class labels are in the last column. Class column may be either a numeric vector or a factor.

Value

A Radviz visualization of the original dataset is returned.

Note

Prior to visualizing, the values of each attribute are usually standardized to the interval [0, 1] to make all the attributes equally important in "pulling" the data point. If one attribute value is much larger than the values of the other attributes, then the point will lie close to the point on the circumference of the circle which corresponds to this attribute. The visualization of a given data set, and also its usefulness, largely depends on the selection of visualized attributes and their ordering around the circle perimeter. The total number of possible orderings of m attributes is factorial(m), but some of them are equivalent up to a rotation or image mirroring. Hence, it can be shown that the total number of different projections with m attributes is factorial(m-1)/2.

Author(s)

Caroline Rodriguez

References

Ankerst M., Keim D. A., Kriegel H.-P. Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets, IEEE Visualization, 1996.

K.A. Olsen, R.R. Korfhage, K.M. Sochats, M.B. Spring and J.G. Williams. Visualisation of a Document Collection: The VIBE System, Information Processing and Management, Vol. 29, No. 1, pp. 69-81, Pergamon Press Ltd, 1993.

See Also

starcoord, surveyplot, parallelplot

Examples

```
data(iris)
radviz2d(iris,"Iris")
```

`rangenorm` *range normalization*

Description

Performs several methods of range normalization.

Usage

```
rangenorm(data, method = c("znorm", "mmnorm", "dscale","signorm", "softnorm"),
superv=TRUE)
```

Arguments

`data`	The name of the dataset to be normalized
`method`	The discretization method to be used:"znorm", "mmnrom", "decscale", "signorm", "softmaxnorm"
`superv`	superv=T for supervised data, that data including the class labels in the last column. if superv=F means that the data to be used is unsupervised.

Details

In the znorm normalization, the mean of each attribute of the transformed set of data points is reduced to zero by subtracting the mean of each attribute from the values of the attributes and dividing the difference by the standard deviation of the attribute. Uses the function scale found in the base library.

Min-max normalization (mmnorm) subtracts the minimum value of an attribute from each value of the attribute and then divides the difference by the range of the attribute. These new values are multiplied by the new range of the attribute and finally added to the new minimum value of the attribute. These operations transform the data into a new range, generally [0,1].

The decscale normalization applies decimal scaling to a matrix or dataframe. Decimal scaling transforms the data into [-1,1] by finding k such that the absolute value of the maximum value of each attribute divided by 10\^k is less than or equal to 1.

In the sigmoidal normalization (signorm) the input data is nonlinearly transformed into [-1,1] using a sigmoid function. The original data is first centered about the mean, and then mapped to the almost linear region of the sigmoid. Is especially appropriate when outlying values are present.

The softmax normalization is so called because it reaches "softly" towards maximum and minimum value, never quite getting there. The transformation is more or less linear in the middle range, and has a nonlinearity at both ends. The output range covered is [0,1]. The algorithm removes the classes of the dataset before normalization and replaces them at the end to form the matrix again.

Value

A matriz containing the discretized data.

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Caroline Rodriguez (2004). An computational environmnent for data preprocessing in supervised classification. Master thesis. UPR-Mayaguez

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#----Several methods of range normalization ----
data(bupa)
bupa.znorm=rangenorm(bupa,method="znorm",superv=TRUE)
bupa.mmnorm=rangenorm(bupa,method="mmnorm",superv=TRUE)
bupa.decs=rangenorm(bupa,method="dscale",superv=TRUE)
bupa.signorm=rangenorm(bupa,method="signorm",superv=TRUE)
bupa.soft=rangenorm(bupa,method="softnorm",superv=TRUE)
#----Plotting to see the effect of the normalization----
op=par(mfrow=c(2,3))
plot(bupa[,1])
plot(bupa.znorm[,1])
plot(bupa.mmnorm[,1])
plot(bupa.decs[,1])
plot(bupa.signorm[,1])
plot(bupa.soft[,1])
par(op)
```

---

reachability *Function for computing the reachability measure in the LOF algortihm*

Description

This function computes the reachability measure for each instance of a dataset. This result is used later to compute the Local Outlyingness Factor.

Usage

```
reachability(distdata, k)
```

Arguments

distdata	The matrix of distances
k	The given number of neighbors

Author(s)

Caroline Rodriguez

See Also

maxlof

---

redundancy *Finding the unique observations in a dataset along with their fequencies*

Description

This function finds out the unique instances in a dataset along with their fequencies.

Usage

```
redundancy(data)
```

Arguments

data	The name of the dataset

Author(s)

Edgar Acuna

See Also

clean

relief *RELIEF Feature Selection*

Description

This function implements the RELIEF feature selection algorithm.

Usage

```
relief(data, nosample, threshold,repet=1)
```

Arguments

data	The dataset for which feature selection will be carried out
nosample	The number of instances drawn from the original dataset
threshold	The cutoff point to select the features
repet	The number of repetitions. It is recommended to use at most 10 repetitions

Details

The general idea of this method is to choose the features that can be most distinguished between classes. These are known as the relevant features. At each step of an iterative process, an instance x is chosen at random from the dataset and the weight for each feature is updated according to the distance of x to its Nearmiss and NearHit. The dataset must have complete cases therefore imputation must be performed in advance.

Value

relevant	A table that gives the ratio between the frequency with which the feature was selected as relevant and the total number of trials performed in one column, and the average weight of the feature in another.
a plot	A plot of the weights of the features

Author(s)

Edgar Acuna

References

KIRA, K. and RENDEL, L. (1992). The Feature Selection Problem : Traditional Methods and a new algorithm. Proc. Tenth National Conference on Artificial Intelligence, MIT Press, 129-134.

KONONENKO, I., SIMEC, E., and ROBNIK-SIKONJA, M. (1997). Overcoming the myopia of induction learning algorithms with RELIEFF. Applied Intelligence Vol7, 1, 39-55.

Examples

```
##---- Feature Selection ---
data(iris)
relief(iris,150,0.01,repet=1)
```

`reliefcat` *Feature selection by the Relief Algorithm for datasets containing nominal features*

Description

This function applies the RELIEF Algorithm to datasets containing nominal attributes.

Usage

```
reliefcat(data, nosample, threshold, vnom, repet)
```

Arguments

`data`	The name of the dataset
`nosample`	The size of the sample drawn and used to update the relevance of each feature. Usually it is equal to the total number of instances.
`threshold`	The threshold for choosing the relevant features
`vnom`	A vector of indices indicating the nominal features
`repet`	The number of the repetitions. It is recommended to use at most 10 repetitions. If the nosample=number of instances then set repet=1

Author(s)

Edgar Acuna

See Also

`relief`

reliefcont *Feature selection by the Relief Algorithm for datasets with only continuous features*

Description

This function applies Relief to datasets containing only continuous attributes.

Usage

```
reliefcont(data, nosample, threshold,repet)
```

Arguments

data	The name of the dataset
nosample	The size of the sample drawn and use to update the relevance of the features
threshold	The threshold for choosing the relevant features.
repet	The number of repetitions. It is recommended to use at most 10 repetitions. When nosample=number of instances use repet=1.

Author(s)

Edgar Acuna

See Also

relief

robout *Outlier Detection with Robust Mahalonobis distance*

Description

This function finds the outliers of a dataset using robust versions of the Mahalanobis distance.

Usage

```
robout(data, nclass=0, meth = c("mve", "mcd"), rep = 10,
plot = TRUE)
```

Arguments

data	The dataset for which outlier detection will be carried out.
nclass	An integer value that represents the class to detect for outliers. By default nclass=0 meaning the column of classes it is not used.
meth	The method used to compute the Mahalanobis distance, "mve"=minimum volume estimator, "mcd"=minimum covariance determinant
rep	Number of repetitions
plot	A boolean value to turn on and off the scatter plot of the Mahalanobis distances

Details

It requires the use of the cov.rob function from the MASS library.

Value

top1	Index of observations identified as top outliers by frequency of selection
topout	Index of observations identified as possible outliers by outlyingness measure
outme	Index of observations and their outlyingness measures

Author(s)

Edgar Acuna

References

Rousseeuw, P, and Leroy, A. (1987). Robust Regression and outlier detection. John Wiley & Sons. New York.

Atkinson, A. (1994). Fast very robust methods for the detection of multiple outliers. Journal of the American Statistical Association, 89:1329-1339.

See Also

robout

Examples

```
## Not run: #---- Outlier Detection in bupa-class 1 using MCD
data(bupa)
robout(bupa,1,"mcd")

## End(Not run)
```

row.matches *Finding rows in a matrix equal to a given vector*

Description

This function finds instances in a data matrix that are equal to a given instance.

Usage

```
row.matches(y, X)
```

Arguments

y	A given instance
X	A given data matrix

Details

This function was found in the CRAN mailing list. It seems to be authored by B. Venables

See Also

redundancy

sbs1 *One-step sequential backward selection*

Description

This functions performs one-step of the sequential backward selection procedure.

Usage

```
sbs1(data, indic, correct0, kvec, method = c("lda", "knn", "rpart"))
```

Arguments

data	The name of a dataset
indic	A vector of 0-1 values: 1 indicates a selected feature.
correct0	The recognition rate based on the current subset of features
kvec	The number of neighbors
method	The classifier to be used

Author(s)

Edgar Acuna

See Also

sffs

---

score *Score function used in Bay's algorithm for outlier detection*

Description

This function finds the score that is used to rank an instance as an outliers.

Usage

```
score(data)
```

Arguments

data	The name of the dataset to be used.

Author(s)

Caroline Rodriguez

See Also

baysout

---

sffs *Sequential Floating Forward Method*

Description

This function selects features using the sequential floating forward method with lda, knn or rpart classifiers.

Usage

```
sffs(data, method = c("lda", "knn", "rpart"), kvec = 5,
repet = 10)
```

Arguments

data	Dataset to be used for feature selection
method	String sequence representing the choice of classifier
kvec	The number of nearest neighbors to be used for the knn classifier
repet	Integer value representing the number of repetitions

Details

The Sequential Floating Forward selection method was introduced to deal with the nesting problem. The best subset of features, T, is initialized as the empty set and at each step a new feature is added. After that, the algorithm searches for features that can be removed from T until the correct classification error does not increase. This algorithm is a combination of the sequential forward and the sequential backward methods. The "best subset" of features is constructed based on the frequency with which each attribute is selected in the number of repetitions given. Due to the time complexity of the algorithm its use is not recommended for data sets with a a large number of attributes(say more than 1000).

Value

fselect	a list of the indices of the best features

Author(s)

Edgar Acuna

References

Pudil, P., Ferri, J., Novovicova, J., and Kittler, J. (1994). Floating search methods for feature selection with nonmonotonic criterion function. 12 International Conference on Pattern Recognition, 279-283.

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Examples

```
#---- SFFS feature selection using the knn classifier ----
data(iris)
sffs(iris,method="rpart",repet=2)
```

sfs *Sequential Forward Selection*

Description

Applies the Sequential Forward Selection algorithm for Feature Selection.

Usage

```
sfs(data, method = c("lda", "knn", "rpart"), kvec = 5,
 repet = 10)
```

Arguments

data	Dataset to be used for feature selection
method	Classifier to be used, currently only the lda, knn and rpart classifiers are supported
kvec	Number of neighbors to use for the knn classification
repet	Number of times to repeat the selection.

Details

The best subset of features, T, is initialized as the empty set and at each step the feature that gives the highest correct classification rate along with the features already in T, is added to set. The "best subset" of features is constructed based on the frequency with which each attribute is selected in the number of repetitions given. Due to the time complexity of the algorithm its use is not recommended for datasets with a large number of attributes(say more than 1000).

Value

bestsubset	subset of features that have been determined to be relevant.

Author(s)

Edgar Acuna

References

Acuna, E , (2003) A comparison of filters and wrappers for feature selection in supervised classification. Proceedings of the Interface 2003 Computing Science and Statistics. Vol 34.

Examples

```
#---- Sequential forward selection using the knn classifier----
data(iris)
sfs(iris,method="lda",repet=3)
```

sfs1 *One-step sequential forward selection*

Description

This function computes one-step of the sequential forward selection procedure.

Usage

```
sfs1(data, indic, correcto, kvec, method = c("lda", "knn",
 "rpart"))
```

Arguments

data	Name of the dataset to be used.
indic	A vector of 0-1 values.
correcto	The recognition rate in the previuos step.
kvec	The number of neighbors to be used by the knn classifier
method	The classifier to be used to select the best features.

Author(s)

Edgar Acuna

See Also

sffs

Shuttle *The Shuttle dataset*

Description

This is the Shuttle dataset from the Stalog project.

Usage

```
data("Shuttle")
```

Format

A data frame with 58000 instances and 10 variables. The shuttle dataset contains 9 attributes all of which are numerical. The last column is the class which has 7 values, 1 Rad Flow 2 Fpv Close 3 Fpv Open 4 High 5 Bypass 6 Bpv Close 7 Bpv Open

Details

Approximately 80

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: data(Shuttle)
mmnorm(Shuttle)

## End(Not run)
```

---

signorm *Sigmoidal Normalization*

Description

Function that performs sigmoidal normalization.

Usage

```
signorm(data)
```

Arguments

data	The dataset to be normalized, including classes

Details

This method transforms the input data nonlinearly into [-1,1] using a sigmoid function. The original data is first centered about the mean, and then mapped to the almost linear region of the sigmoid. Is especially appropriate when outlying values are present.

Removes classes before normalization, and returns the normalized data set complete with classes rejoined.

Value

sigdata	Original dataset normalized

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
#---- Sigmoidal Normalization ---
data(vehicle)
vehicle.signorm=signorm(vehicle)
op=par(mfrow=c(2,1))
plot(vehicle[,1])
plot(vehicle.signorm[,1])
par(op)
```

`softmaxnorm` *Softmax Normalization*

Description

This is a function that applies softmax normalization to a matrix or dataframe.

Usage

```
softmaxnorm(data)
```

Arguments

`data`	The datset to be normalized

Details

This normalization is so called because it reaches "softly" towards maximum and minimum value, never quite getting there. The transformation is more or less linear in the middle range, and has a nonlinearity at both ends. The output range covered is [0,1]. The algorithm removes the classes of the dataset before normalization and replaces them at the end to form the matrix again.

Value

`softdata`	original matrix normalized

Author(s)

Caroline Rodriguez and Edgar Acuna

Examples

```
#---- Softmax Normalization----
data(sonar)
sonar.sftnorm=softmaxnorm(sonar)
op=par(mfrow=c(2,1))
plot(sonar[,1])
plot(sonar.sftnorm[,1])
par(op)
```

---

sonar *The Sonar dataset*

Description

This is the sonar dataset. It containis information on 208 objects and 60 attributes. The objects are classified in two classes: "rock" and "mine".

Usage

```
data(sonar)
```

Format

A data frame with 208 observations on 61 variables. The first 60 represent the energy within a particular frequency band,integrated over a certain period of time. The last column contains the class labels. There are two classes 0 if the object is a rock, and 1 if the object is a mine (metal cylinder). The range value of each attribute varies from 0.0 to 1.0.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
## Not run: #Robust detection of outliers in sonar-class1 using MVE----
data(sonar)
robout(sonar,1,"mve",rep=10)

## End(Not run)
```

srbct *Khan et al.'s small round blood cells dataset*

Description

The sbrct dataset which contains information on 63 samples and 2308 genes. The samples are distributed in four classes as follows: 8 Burkitt Lymphoma (BL), 23 Ewing Sarcoma (EWS), 12 neuroblastoma (NB), and 20 rhabdomyosarcoma (RMS). The last column contains the class labels.

Usage

```
data(srbct)
```

Format

A data frame containing 63 observations with 2308 attributes each. The last column of the dat frame contains the class labels for each observation.

Source

The data set was obtained, as binary R file from Marcel Dettling's web site:

- http://stat.ethz.ch/~dettling/bagboost.html

References

Javed Khan, Jun S. Wei, Markus Ringner, Lao H. Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson, and Paul S. Meltzer (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine, Volume 7, Number 6, June

Examples

```
#---z-score Normalization
data(srbct)
srbct.rnorm=rangenorm(srbct,"znorm")
#---feature selection using the RELIEF feature selection algorithm-----
#relief(srbct,63,0.12)
```

star3d *Data Visuaization using star coordinates in three dimensions*

Description

This function perform data visulization using cooordinates in three dimensions. Rotation and zooming is possible calling to the rgl library

Usage

```
star3d(data, vars = NULL, scale = 1)
```

Arguments

data	The dataset to be visualized
vars	The variables to be scaled
scale	the scale factor

Author(s)

Edgar Acuna

See Also

starcoord

Examples

```
## Not run:
data(vehicle)
star3d(vehicle)

## End(Not run)
```

starcoord *The star coordinates plot*

Description

This function displays a star coordinates plot introduced by Kondogan (2001).

Usage

```
starcoord(data, main = NULL, class = FALSE, outliers=NULL, vars = 0,
scale = 1, cex = 0.8, lwd = 0.25, lty = par("lty"))
```

Arguments

data	The dataset
main	The title of the plot
class	This logical variable is TRUE for supervised data and FALSE for unsupervised data
outliers	The instances to be highlighted as potential outliers
vars	The variables to be scaled
scale	The scale factor
cex	A numerical value giving the amount by which plotting text and symbols should be scaled.
lwd	The width of the lines representing the axis
lty	The type of the lines representing the axis

Details

This plot is not recommended for a large number of features (say more than 50).

Value

Returns a Star Coordinates Plot of the data matrix

Author(s)

Edgar Acuna and Shiyun Wen

References

E. Kandogan (2001). Visualizing multidimensional clusters, Trends, and Outliers, using star coordinates. Proceedings of KDD 2001.

See Also

parallelplot, surveyplot

Examples

```
data(vehicle)
starcoord(vehicle, main="Vehicle Dataset", class=TRUE, outliers=NULL,vars=0, scale=1,
cex=0.8, lwd = 0.25, lty = par("lty"))
```

surveyplot *Surveyplot*

Description

This function creates and displays a surveyplot of a dataset for a classification matrix

Usage

```
surveyplot(datos, dataname = "", orderon = 0, class = 0,
obs = rep(0, 0), lwd = 1)
```

Arguments

datos	A matrix of values for supervised classification
dataname	dataname Name of data set to appear in plot title
orderon	orderon Column number by which to order the dataset
class	class Class for which to limit plotting
obs	obs List of observations to be highlighted
lwd	lwd Value to control width of the line

Details

This plot is not recommended for a large number of features (say more than 50)

Value

Returns a surveyplot of the data matrix

Note

This plot is a mix between the survey plot presented in Fayyad and a permutation matrix.

Author(s)

Caroline Rodriguez

References

Fayyad, et al. (2001) Information Visualization in Data Mining and Knowledge Discovery

See Also

parallelplot, starcoord

Examples

```
#----Surveyplot examples
data(bupa)
surveyplot(bupa,"Bupa Dataset")
surveyplot(bupa,"Bupa Dataset",orderon=1,obs=c(6,74,121))
```

`tchisq` *Auxiliary function for the Chi-Merge discretization*

Description

This function is required to compute the chi-Merge discretization.

Usage

```
tchisq(obs)
```

Arguments

`obs`	a vector of observed frequencies

Author(s)

Jaime Porras

See Also

`chiMerge`

`top` *Auxiliary function for Bay's Ouylier Detection Algorithm*

Description

Function that finds the number of candidate outliers requested by the user.

Usage

```
top(O, neighbors, n)
```

Arguments

`O`	An n x 1 matrix with the score function from k nearest neighbors
`neighbors`	The number of neighbors to be considered
`n`	The number of top outliers to search for.

Author(s)

Caroline Rodriguez

See Also

baysout

---

unor *Auxiliary function for performing Holte's 1R discretization*

Description

This function is called by the disc.1r function

Usage

```
unor(a, binsize, out = c("symb", "num"))
```

Arguments

a	a is a two column matrix where the first column contains the values to be discretized and the second column contains the class labels.
binsize	the minimum number of attributes values in each bin.
out	To get the discretized data in numerical format enter "nun". To get the discretized data in interval format enter "symb".

Value

Returns the discretized values of the first column of the matrix a.

Author(s)

Edgar Acuna

See Also

disc.1r

`vehicle` *The Vehicle dataset*

Description

This is the Vehicle dataset from the UCI Machine Learning Repository

Usage

```
data(vehicle)
```

Format

A data frame with 846 observations on the following 19 variables.

V1 Compactness

V2 Circularity

V3 Distance Circularity

V4 Radius ratio

V5 pr.axis aspect ratio

V6 max.length aspect ratio

V7 scatter ratio

V8 elongatedness

V9 pr.axis rectangularity

V10 max.length rectangularity

V11 scaled variance along major axis

V12 scaled variance along minor axis

V13 scaled radius of gyration

V14 skewness about major axis

V15 skewness about minor axis

V16 kurtosis about minor axis

V17 kurtosis about major axis

V18 hollows ratio

V19 Type of vehicle: a double decker bus, Cheverolet van, Saab 9000 and an Opel Manta 400.

Source

The UCI Machine Learning Database Repository at:

- ftp://ftp.ics.uci.edu/pub/machine-learning-databases
- http://www.ics.uci.edu/~mlearn/MLRepository.html

Examples

```
#----feature selection using sequential floating selection with LDA----
data(vehicle)
mahaout(vehicle,nclass=3)
```

---

vvalen *The Van Valen test for equal covariance matrices*

Description

The Van Valen nonparametric test for homocedasticity (equal covariance matrices).

Usage

```
vvalen(data)
```

Arguments

data	The name of the dataset to be tested

Value

Gives the p-value for a Kruskal Wallis test. A p-value near to zero indicates homocedasticity.

Author(s)

Edgar Acuna

References

Van Valen, L. (1962). A study of fluctuating asymmetry. Evolution Vol. 16, pp. 125142.

See Also

mardia

Examples

```
#---------Testing homocedasticity-----
data(colon)
vvalen(colon)
```

vvalen1 *Auxiliary function for computing the Van Valen's homocedasticity test*

Description

This function is required to perform the Van Valen's homocedasticty test.

Usage

```
vvalen1(data, classn)
```

Arguments

data	The name of the dataset to be considered
classn	The class numnber

Author(s)

Edgar Acuna

See Also

vvalen

znorm *Z-score normalization*

Description

This is a function to apply z-Score normalization to a matrix or dataframe.

Usage

```
znorm(data)
```

Arguments

data	The dataset to be normalized, including classes

Details

By using this type of normalization, the mean of the transformed set of data points is reduced to zero by subtracting the mean of each attribute from the values of the attributes and dividing the result by the standard deviation of the attribute. Uses the function scale found in the base library.

Removes classes before normalization, and returns normalized data set complete with classes re-joined.

Value

zdata	the normalized data set

Author(s)

Caroline Rodriguez and Edgar Acuna

References

Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

Examples

```
##---- Z-norm normalization ----
data(diabetes)
diab.znorm=znorm(diabetes)
op=par(mfrow=c(2,1))
plot(diabetes[,1])
plot(diab.znorm[,1])
par(op)
```

Index