Package 'pdfCluster'

December 2, 2022

Type Package

Title Cluster Analysis via Nonparametric Density Estimation

Version 1.0-4

Date 2022-12-01

Maintainer Menardi Giovanna <menardi@stat.unipd.it>

Description Cluster analysis via nonparametric density estimation is performed. Operationally, the kernel method is used throughout to estimate the density. Diagnostics methods for evaluating the quality of the clustering are available. The package includes also a routine to estimate the probability density function obtained by the kernel method, given a set of data with arbitrary dimensions.

License GPL-2

LazyLoad yes

Imports geometry, methods

Suggests cluster

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-12-02 09:50:02 UTC

Encoding UTF-8

Language en-US

Author Menardi Giovanna [aut, cre],
Azzalini Adelchi [aut],
Rosolin Tiziana [ctb] (up to version 0.1-13)

R topics documented:

`pdfCluster-package` *The pdfCluster package: summary information*

Description

This package performs cluster analysis via kernel density estimation (Azzalini and Torelli, 2007; Menardi and Azzalini, 2014). Clusters are associated to the maximally connected components with estimated density above a threshold. As the threshold varies, these clusters may be represented according to a hierarchical structure in the form of a tree. Detection of the connected regions is conducted by means of the Delaunay tesselation when data dimensionality is low to moderate, following Azzalini and Torelli (2007). For higher dimensional data, detection of connected regions is performed according to the procedure described in Menardi and Azzalini (2013). In both cases, after that a number of high-density cluster-cores is identified, lower density data are allocated by following a supervised classification-like approach. The number of clusters, corresponding to the number of the modes of the estimated density, is automatically selected by the procedure. Diagnostics methods for evaluating the quality of clustering are also available (Menardi, 2011). Moreover, the package provides a routine to estimate the probability density function by kernel methods, given a set of data with arbitrary dimension. The main features of the package are described and illustrated in Azzalini and Menardi (2014).

Details

The `pdfCluster-package` makes use of classes and methods of the S4 system. It includes some foreign functions written in the C language: two of them compute the kernel density estimate of data and are interfaced by the R function `kepdf`. Other C routines included in the package allow for a quicker detection of the connected components of the subgraphs associated with the level sets of the data. Two of them are directly drawn from the homonymous ones in the `spdep` package.

Starting from version 1.0-0, new features have been introduced:

- kernel density estimation may be performed by using either a a fixed or an adaptive bandwidth; moreover, the option of selecting a Student's t kernel has been included, for computational convenience;
- detection of connected components of the level sets is performed by means of the Delaunay triangulation when data dimensionality is up to 6, following Azzalini and Torelli (2007); for higher dimensional data a new procedure, which is less time-consuming, is now adopted (Menardi and Azzalini, 2014);
- the order of classification of lower density data depends now also on the standard error of the estimated density ratios; moreover, a cluster-specific bandwidth is the default option to classify low density data.

See examples below to understand how to set arguments of the main function of the package, in order to obtain the same results as the ones obtained with versions 0.1-x.

Author(s)

Adelchi Azzalini, Giovanna Menardi, Tiziana Rosolin

Maintainer: Giovanna Menardi <menardi at stat.unipd.it>

References

Azzalini, A., Menardi, G. (2014). Clustering via Nonparametric Density Estimation: The R Package pdfCluster. *Journal of Statistical Software*, 57(11), 1-26, URL http://www.jstatsoft.org/v57/i11/.

Azzalini A., Torelli N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*, 17, 71-80.

Menardi G. (2011). Density based Silhouette diagnostics for clustering methods. *Statistics and Computing*, 21, 295-308.

Menardi G., Azzalini, A. (2014). An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, DOI: 10.1007/s11222-013-9400-x, to appear.

Examples

```
# load data
data(wine)
gr <- wine[, 1]

# select a subset of variables
x <- wine[, c(2, 5, 8)]

#density estimation
pdf <- kepdf(x)
summary(pdf)
plot(pdf)

#clustering
cl <- pdfCluster(x)
summary(cl)
plot(cl)
```

```
#comparison with original groups
table(groups(cl),gr)

#density based silhouette diagnostics
dsil <- dbs(cl)
plot(dsil)

##########
# higher dimensions

x <- wine[, -1]

#density estimation with adaptive bandwidth
pdf <- kepdf(x, bwtype="adaptive")
summary(pdf)
#density plot is not much clear for high- dimensional data
#select a few variables
plot(pdf, indcol = c(1,4,7))

#clustering
#when dimension is >= 6, default method to find connected components is "pairs"
#density is better estimated by using an adaptive bandwidth
cl <- pdfCluster(x, bwtype="adaptive")
summary(cl)
plot(cl)

########
# this example shows how to set the arguments in function pdfCluster
# in order to obtain the same results as the ones of versions 0.1-x.
x <- wine[, c(2, 5, 8)]

# previous versions of the package
# do not run
# old code:
# cl <- pdfCluster(x)

# same result is obtained now obtained as follows:
cl <- pdfCluster(x, se=FALSE, hcores= TRUE, graphtype="delaunay", n.grid=50)
```

`adj.rand.index` *Adjusted Rand index*

Description

Computes the adjusted Rand index to compare two alternative partitions of the same set.

Usage

```
adj.rand.index(cl1, cl2)
```

Arguments

cl1	the vector containining the class labels of the first partition.
cl2	the vector containining the class labels of the second partition.

Details

The adjusted Rand index is a correction of the Rand index that measures the similarity between two classifications of the same objects by the proportions of agreements between the two partitions. The correction is obtained by subtracting from the Rand index its expected value.

Value

A numeric vector of length 1.

References

L. Hubert and P. Arabie (1985) Comparing partitions, *Journal of Classification*, 2, 193-218.

See Also

table

Examples

```
# load data
data(wine)
#actual groups
gr <- wine[, 1]

# select a subset of variables
x <- wine[, c(2, 5, 8)]

#clustering
cl <- pdfCluster(x)

#comparison with original groups
table(groups(cl), gr)
adj.rand.index(groups(cl), gr)
```

---

dbs *Density-based silhouette information methods*

Description

Computes the density-based silhouette information of clustered data. Two methods are associated to this function. The first method applies to two arguments: the matrix of data and the vector of cluster labels; the second method applies to objects of pdfCluster-class.

Usage

```
## S4 method for signature 'matrix'
dbs(x, clusters, h.funct="h.norm", hmult=1, prior, ...)

## S4 method for signature 'pdfCluster'
dbs(x, h.funct="h.norm", hmult = 1, prior =
   as.vector(table(x@cluster.cores)/sum(table(x@cluster.cores))),
   stage=NULL, ...)
```

Arguments

x	A matrix of data points partitioned by any density-based clustering method or an object of pdfCluster-class.
clusters	Cluster labels of grouped data. This argument has not to be set when x is a pdfCluster-class object.
h.funct	Function to estimate the smoothing parameters. Default is h.norm.
hmult	Shrink factor to be multiplied by the smoothing parameters. Default value is 1.
prior	Vector of prior probabilities of belonging to the groups. When x is of pdfCluster-class, default value is set proportional to the cluster cores cardinalities. Otherwise, equal prior probabilities are given to the clusters by default.
stage	When x is a pdfCluster-class object, this is the stage of classification of low-density data at which the dbs has to be computed. Default value is the number of stages of the procedure. Set it to 0 if the dbs has to be computed at cluster cores only.
...	Further arguments to be passed to methods (see dbs-methods) or arguments to kepdf. See details below.

Details

This function provides diagnostics for a clustering produced by any density-based clustering method. The dbs information is a suitable modification of the silhouette information aimed at evaluating the cluster quality in a density based framework. It is based on the estimation of data posterior probabilities of belonging to the clusters. It may be used to measure the quality of data allocation to the clusters. High values of the $\hat{dbs}$ are evidence of a good quality clustering.

Define

$$\hat{\tau}_m(x_i) = \frac{\pi_m \hat{f}(x_i|x_{\in} m)}{\sum_{m=1}^{M} \pi_m \hat{f}(x_i|x_i \in m)} \quad m = 1, \ldots, M,$$

where π_m is a prior probability of m and $\hat{f}(x_i|x_i \in m)$ is a density estimate at x_i evaluated with function kepdf by using the only data points in m. Density estimation is performed with fixed bandwidths h, as evaluated by function h.funct, possibly multiplied by the shrink factor hmult.

Density-based silhouette information of x_i, the i^{th} row of the data matrix x, is defined as follows:

$$\hat{dbs}_i = \frac{\log\left(\frac{\hat{\tau}_{m_0}(x_i)}{\hat{\tau}_{m_1}(x_i)}\right)}{\max_{x_i}\left|\log\left(\frac{\hat{\tau}_{m_0}(x_i)}{\hat{\tau}_{m_1}(x_i)}\right)\right|},$$

where m_0 is the group where x_i has been allocated and m_1 is the group for which τ_m is maximum, $m \neq m_0$.

Note: when there exists x_j such that $\hat{\tau}_{m_1}(x_j)$ is zero, $\hat{dbs}_j$ is forced to 1 and $\max_{x_i} \left| \log \left(\frac{\hat{\tau}_{m_0}(x_i)}{\hat{\tau}_{m_1}(x_i)} \right) \right|$ is computed by excluding x_j from the data matrix `x`.

See Menardi (2011) for a detailed treatment.

Value

An object of class `"dbs"`, with slots:

`call`	The matched call.
`x`	The matrix of clustered data points.
`prior`	The vector of prior probabilities of belonging to the groups.
`dbs`	A vector reporting the density-based silhouette information of the clustered data.
`clusters`	Cluster labels of grouped data.
`noc`	Number of clusters
`stage`	If argument `x` of `dbs` is a `pdfCluster-class` object, this slot provides the stage of the classification at which the dbs is computed.

See `dbs-class` for more details.

Methods

`signature(x = "matrix", clusters = "numeric")` Computes the density based silhouette information for objects partitioned according to any density-based clustering method.

`signature(x = "pdfCluster", clusters = "missing")` Computes the density based silhouette information for objects of class `"pdfCluster"`.

References

Menardi, G. (2011) Density-based Silhouette diagnostics for clustering methods. *Statistics and Computing*, 21, 295-308.

See Also

`dbs-class`, `plot,dbs-method`, `silhouette`.

Examples

```
#example 1: no groups in data
#random generation of group labels
set.seed(54321)
x <- rnorm(50)
groups <- sample(1:2, 50, replace = TRUE)
groups
dsil <- dbs(x = as.matrix(x), clusters=groups)
dsil
summary(dsil)
```

```
plot(dsil, labels=TRUE, lwd=6)

#example 2: wines data
# load data
data(wine)

# select a subset of variables
x <- wine[, c(2,5,8)]

#clustering
cl <- pdfCluster(x)

dsil <- dbs(cl)
plot(dsil)
```

---

dbs-class *Class "dbs"*

Description

This class pertains to results of the application of function dbs.

Objects from the Class

Objects can be created by calls of the form `new("dbs", ...)` or as a result from calling function dbs.

Slots

call: Object of class "call" reporting the matched call.

x: Object of class "matrix" representing the clustered data points.

prior: Object of class "numeric" being the prior probabilities of belonging to the groups.

dbs: Object of class "numeric" reporting the density-based silhouette information of the clustered data.

clusters: Object of class "numeric" reporting the group labels of grouped data.

noc: Object of class "numeric" indicating the number of clusters.

stage: Object of class "ANY" corresponding to the stage of the classification at which the density-based silhouette information is computed when dbs is applied to an object of pdfCluster-class.

Methods

plot `signature(x = "dbs", y = "missing")`:

S4 method for plotting objects of dbs-class. Data are partitioned into the clusters, sorted in a decreasing order with respect to their dbs value and displayed on a bar graph. See plot,dbs-method for further details.

show `signature(object = "dbs")`:

S4 method for showing objects of dbs-class. The following elements are shown:

- the dbs index computed at the observed data;
- The cluster membership of each data point;

summary `signature(object = "dbs")`:

S4 method for summarizing objects of dbs-class. The following elements are shown:

- a summary (minimum, 1st quartile, median, mean, 3rd quartile, maximum) of the dbs values for each cluster;
- a summary (minimum, 1st quartile, median, mean, 3rd quartile, maximum) of the dbs values for all the observations.

See Also

dbs, silhouette, plot,dbs-method, plot-methods, show-methods, summary-methods.

Examples

```
showClass("dbs")

#wine example
#data loading
data(wine)

# select a subset of variables
x <- wine[, c(2,5,8)]

#clustering
cl <- pdfCluster(x)

dsil <- dbs(cl)
dsil
summary(dsil)
```

`groups` *Extracts groups*

Description

Extracts the detected groups from objects of pdfCluster-class.

Usage

```
groups(obj, stage = length(obj@stages))
```

Arguments

obj	An object of pdfCluster-class
stage	The stage of classification at which the clusters have to be extracted. Set this value to 0 to extract the cluster cores. Default value is the total number of classification stages, that is, the final partition is given. When obj contains the clusters cores only, these are given by default.

Details

This function is an user-friendly version of command obj@clusters, now obsolete, to ease extraction of groups from objects of pdfCluster-class.

Value

A numeric vector containing the group labels. NA values are associated to points not classified at the selected stage of the classification procedure.

See Also

pdfCluster

Examples

```
# load data
data(wine)

# select a subset of variables
x <- wine[, c(2, 5, 8)]

#clustering
cl <- pdfCluster(x)

groups(cl)

#equivalent to:
cl@clusters

#to extract the cluster cores
groups(cl, stage = 0)
```

---

h.norm *Normal optimal choice of smoothing parameter in density estimation*

Description

This function computes the smoothing parameter to be used in kernel density estimation, as asymptotically optimal when the underlying distribution is Normal. Unidimensional as well as multidimensional data can be handled. When multidimensional data are supplied, a vector of smoothing parameters is computed having one element for each component.

Usage

```
h.norm(x)
```

Arguments

x	vector, matrix or data-frame of data.

Details

The smoothing parameter of component j of a $n \times d$ data matrix is estimated as follows:

$$\sigma_j \left(\frac{4}{(d+2)n} \right)^{\frac{1}{d+4}}$$

where σ_j is the estimated standard deviation of component j. See Section 2.4.2 of the reference below.

Value

Returns a numeric vector with the same length as the number of columns of x or with length one if x is a vector. When x is a matrix, a vector of smoothing parameters is provided having one element for each component.

References

Bowman, A.W. and Azzalini, A. (1997). *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Oxford University Press, Oxford.

See Also

hnorm

Examples

```
set.seed(123)
x <- rnorm(30)
sm.par <- h.norm(x)
pdf <- kepdf(x, bwtype= "fixed", h = sm.par)
plot(pdf, eval.points=seq(-4,4,by=.2))
```

hprop2f *Sample smoothing parameters in adaptive density estimation*

Description

This function computes the sample smoothing parameters to be used in adaptive kernel density estimation, according to Silverman (1986).

Usage

```
hprop2f(x, h = h.norm(x), alpha = 1/2, kernel = "gaussian")
```

Arguments

x	Vector or matrix of data.
h	Vector of smoothing parameters to be used to get a pilot estimate of the density function. It has length equal to NCOL(x).
alpha	Sensitivity parameter satysfying $0 \leq \alpha \leq 1$, giving the power to which raise the pilot density. Default value is 1/2. See details.
kernel	Kernel to be used to compute the pilot density estimate. It should be one of "gaussian" or "t7". See kepdf for further details.

Details

A vector of smoothing parameters h_i is chosen for each sample point x_i, as follows:

$$h_i = h \left(\frac{\hat{f}_h(x_i)}{g} \right)^{-\alpha}$$

where $\hat{f}_h$ is a pilot kernel density estimate of the density function f, with vector of bandwidths h, and g is the geometric mean of $\hat{f}_h(x_i)$, $i = 1, ..., n$. See Section 5.3.1 of the reference below.

Value

Returns a matrix with the same dimensions of x where row i provides the vector of smoothing parameters for sample point x_i.

References

Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall, London.

See Also

h.norm

Examples

```
set.seed(123)
x <- rnorm(10)

sm.par <- hprop2f(x)
pdf <- kepdf(x, bwtype= "adaptive")

pdf@par$hx
sm.par

plot(pdf,eval.points=seq(-4,4,by=.2))
```

---

kepdf *Kernel estimate of a probability density function.*

Description

Estimates density of uni- and multivariate data by the kernel method.

Usage

```
kepdf(x, eval.points = x, kernel = "gaussian",
      bwtype = "fixed", h = h.norm(x), hx = NULL, alpha = 1/2)
```

Arguments

x	A vector, a matrix or data-frame of data whose density should be estimated.
eval.points	A vector, a matrix or a data-frame of data points at which the density estimate should be evaluated.
kernel	Either 'gaussian' or 't7', it defines the kernel function to be used. See details below.
bwtype	Either 'fixed' or 'adaptive', corresponding to a kernel estimator with fixed or adaptive bandwidths respectively. See details below.
h	A vector of length set to NCOL(x), defining the smoothing parameters to be used either to estimate the density in kernel estimation with fixed bandwidth or to estimate the pilot density in kernel estimation with adaptive bandwidths. Default value is the result of h.norm applied to x.
hx	A matrix with the same number of rows and columns as x, where each row defines the vector of smoothing parameters specific for each sample point. To be used when bwtype = "adaptive". Default value is the result of hprop2f applied to x. Set to NULL when bwtype= "fixed".
alpha	Sensitivity parameter to be given to hprop2f when bwtype= "adaptive" and the vectors of smoothing parameters are computed according to Silverman's (1986) approach.

Details

The current version of pdfCluster-package allows for computing estimates by a kernel product estimator of the form:

$$\hat{f}(y) = \sum_{i=1}^{n} \frac{1}{n h_{i,1} \cdots h_{i,d}} \prod_{j=1}^{d} K\left(\frac{y_j - x_{i,j}}{h_{i,j}}\right).$$

The kernel function K can either be a Gaussian density (if `kernel = "gaussian"`) or a t_ν density, with $\nu = 7$ degrees of freedom (when `kernel = "t7"`). Although uncommon, the option of selecting a t kernel is motivated by computational efficiency reasons. Hence, its use is suggested when either `x` or `eval.points` have a huge number of rows.

The vectors of bandwidths $h_i = (h_{i,1} \cdots h_{i,d})'$ are defined as follows:

Fixed bandwidth When `bwtype='fixed'`, $h_i = h$ that is, a constant smoothing vector is used for all the observations x_i. Default values are set as asymptotically optimal for a multivariate Normal distribution (e.g., Bowman and Azzalini, 1997). See `h.norm` for further details.

Adaptive bandwidth When `bwtype='adaptive'`, a vector of bandwidths h_i is specified for each observation x_i. Default values are selected according to Silverman (1986, Section 5.3.1). See `hprop2f`.

Value

An S4 object of `kepdf-class` with slots:

`call`	The matched call.
`x`	The data input, coerced to be a matrix.
`eval.points`	The data points at which the density is evaluated.
`estimate`	The values of the density estimate at the evaluation points.
`kernel`	The selected kernel.
`bwtype`	The type of estimator.
`par`	A list of parameters used to estimate the density, with elements: • `h` the smoothing parameters used to estimate either the density or the pilot density; • `hx` the matrix of sample smoothing parameters, when `bwtype='adaptive'`; • `alpha` sensitivity parameter used if `bwtype='adaptive'`.

References

Bowman, A.W. and Azzalini, A. (1997). *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Oxford University Press, Oxford.

Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall, London.

See Also

`h.norm`, `hprop2f`, `kepdf-class`.

Examples

```
## A 1-dimensional example
data(wine)
x <- wine[,3]
pdf <- kepdf(x, eval.points=seq(0,7,by=.1))
plot(pdf, n.grid= 100, main="wine data")

## A 2-dimensional example
x <- wine[,c(2,8)]
pdf <- kepdf(x)
plot(pdf, main="wine data", props=c(5,50,90), ylim=c(0,4))
plot(pdf, main="wine data", method="perspective", phi=30, theta=60)

### A 3-dimensional example
x <- wine[,c(2,3,8)]
pdf <- kepdf(x)
plot(pdf, main="wine data", props=c(10,50,70), gap=0.2)
plot(pdf, main="wine data", method="perspective", gap=0.2, phi=30, theta=10)

### A 6-dimensional example
### adaptive kernel density estimate is preferable in high-dimensions
x <- wine[,c(2,3,5,7,8,10)]
pdf <- kepdf(x, bwtype="adaptive")
plot(pdf, main="wine data", props=c(10,50,70), gap=0.2)
plot(pdf, main="wine data", method="perspective", gap=0.2, phi=30, theta=10)
```

`kepdf-class` *Class "kepdf"*

Description

This class encapsulates results of the application of function `kepdf`.

Objects from the Class

Objects can be created by calls of the form `new("kepdf", ...)` or as a result of a call to `kepdf`.

Slots

`call`: Object of class `"call"`, corresponding to the matched call.

`x`: Object of class `"matrix"` representing the data points used to estimate the probability density function.

`eval.points`: Object of class `"matrix"` representing the data points at which the density is evaluated.

estimate The values of the density estimate at the evaluation points.

`kernel`: Object of class `"character"` giving the selected kernel.

`bwtype`: Object of class `"character"` giving the selected type of estimator.

`par`: Object of class `"list"` providing the parameters used to estimate the density. Its elements are `h`, `hx`, and possibly `alpha`.

See `kepdf` for further details.

Methods

plot `signature(x = "kepdf", y = "ANY")`

Plots objects of `kepdf-class`. `plot-methods` are available for density estimates of:

- one-dimensional data;
- two-dimensional data: contour, image or perspective plots are available;
- multi-dimensional data: matrix of plots of all the pairs of two-dimensional marginal kernel density estimates.

See `plot,kepdf-method` for further details.

show `signature(object = "kepdf")`

Prints the following elements:

- the class of the object;
- the selected kernel;
- the selected type of estimator;
- either the fixed smoothing parameters or the smoothing parameters of each observation;
- the density estimates at the evaluation points.

summary `signature(object = "kepdf")`

Provides a summary of `kepdf-class` object by printing the highest density data point and the row or index position of a percentage top density data points, possibly given as optional argument `prop`.

See Also

`h.norm`, `kepdf`, `plot,kepdf-method`, `plot-methods`, `show-methods`, `summary-methods`.

Examples

```
#
showClass("kepdf")

#
data(wine)
#select only "Barolo"-type wines
x <- wine[1:59,3]
pdf <- kepdf(x)
pdf
summary(pdf)
summary(pdf, props = 10*seq(1, 9, by = 1))
```

`oliveoil` *Olive oil data*

Description

This data set represents eight chemical measurements on different specimen of olive oil produced in various regions in Italy (northern Apulia, southern Apulia, Calabria, Sicily, inland Sardinia and coast Sardinia, eastern and western Liguria, Umbria) and further classifiable into three macro-areas: Centre-North, South, Sardinia. The data set is used to evaluate the `pdfCluster` ability of recunstructing the macro-area membership.

Usage

```
data(oliveoil)
```

Format

This data frame contains 572 rows, each corresponding to a different specimen of olive oil, and 10 columns. The first and the second column correspond to the macro-area and the region of origin of the olive oils respectively; here, the term "region" refers to a geographical area and only partially to administrative borders. Columns 3-10 represent the following eight chemical measurements on the acid components for the oil specimens: palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, eicosenoic.

Details

Since the raw data are of compositional nature, ideally totalling 10000, some preliminary transformations of data are advisable. In particular, Azzalini and Torelli (2007) adopt an additive log-ratio transformation (ALR). If x_j denotes the j^{th} chemical measurement ($j = 1, \ldots, 8$), the ALR transformation is $y_j = \log x_j/x_k, j \neq k$, where k is an arbitrary but fixed variable. However, in this data set, the raw data do not always sum up exactly to 10000, because of measurement errors. Moreover, some 0's are present in the data, corresponding to measurements below the instrument sensitivity level. Therefore, it is suggested to add 1 to all raw data and normalize them by dividing each entry by the corresponding row sum $\sum_j (x_j + 1)$.

Source

Forina, M., Lanteri, S. Armanino, C., Casolino, C., Casale, M., Oliveri, P. (2008). V-PARVUS. *An Extendible Package of programs for explorative data analysis, classification and regression analysis*. Dip. Chimica e Tecnologie Farmaceutiche ed Alimentari, Università di Genova.

References

Azzalini A., Torelli N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*, 17, 71-80.

`pdfClassification` *Classification of low density data*

Description

Allocates low density data points in a multi-stage procedure after that cluster cores have been detected by applying `pdfCluster`.

Usage

```
pdfClassification(obj, n.stage = 5, se = TRUE, hcores = FALSE)
```

Arguments

`obj`	An object of `pdfCluster-class`.
`n.stage`	Allocation of low density data is performed by following a multi-stages procedure in `n.stage` stages.
`se`	Logical. Should the standard-error of the density estimates be taken into account to define the order of allocation? Default value is TRUE. See details below.
`hcores`	Logical. Set this value to TRUE to build cluster density estimates by selecting the same bandwidths as the ones used to form the cluster cores. Otherwise, bandwidths specific for the clusters are selected. Default value is FALSE. See details below.

Details

The basic idea of the classification stage of the procedure is as follows: for an unallocated data point x_0, compute the estimated density $\hat{f}_m(x_0)$ based on the data already assigned to group $m, m = 1, 2, \ldots, M$, and assign x_0 to the group with highest log ratio $\hat{f}_m(x_0)/\max_m \hat{f}_m(x_0)$.

In case $\hat{f}_m(x_0)$=0, for all $m = 1, 2, \ldots, M$, x_0 is considered as an outlier. The procedure gives a warning message and the outlier remains unclassified. The cluster label of x_0 will be set to zero.

The current implementation of this idea proceeds in `n.stage` stages, allocating a block of points at a time, updating the estimates $\hat{f}_m(\cdot)$ based on the new members of each group and then allocating a new block of points. When `se = TRUE`, classification is performed by further weighting the log-ratios inversely with their approximated standard error, so that points whose density estimate has highest precision are allocated first.

Each of the $\hat{f}_m(\cdot)$ is built by selecting either the same bandwidths h_0 as the ones used to form the cluster cores (when `hcores = TRUE`) or cluster-specific bandwidths, obtained as follows:

$$h_m^* = \exp[(1 - a_m)\log(h_0) + a_m \log(h_m)],$$

where a_m is the proportion of data points in the m-th cluster core and h_m are asymptotically optimal for a normal distribution of the m-th cluster or computed according to the Silverman (1986) approach, if the kernel estimator has fixed or adaptive bandwidth, respectively.

Value

An object of pdfCluster-class with slot stages of class "list" having length equal to n.stage. See pdfCluster-class for further details.

Note

Function pdfClassification is called internally, from pdfCluster, when the argument n.stage is set to a value greater than zero. Alternatively, it may be called externally, by providing as argument an object of pdfCluster-class.

When pdfClassification is internally called from pdfCluster and one group only is detected, the slot stages is a list with n.stage elements, each of them being a vector with length equal to the number of data points and all elements equal to 1.

References

Azzalini A., Torelli N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*. 17, 71-80.

Silverman, B. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall, London.

See Also

pdfCluster, pdfCluster-class

Examples

```
# load data
data(wine)

# select a subset of variables
x <- wine[, c(2,5,8)]

#whole procedure, included the classification phase
cl <- pdfCluster(x)
summary(cl)
table(groups(cl))

#use of bandwidths specific for the group
cl1 <- pdfClassification(cl, hcores= TRUE)
table(groups(cl1))
```

---

pdfCluster *Clustering via nonparametric density estimation*

Description

Cluster analysis is performed by the density-based procedures described in Azzalini and Torelli (2007) and Menardi and Azzalini (2014), and summarized in Azzalini and Menardi (2014).

Usage

```
## S4 method for signature 'numeric'
pdfCluster(x, graphtype, hmult, Q = "QJ", lambda = 0.1,
   grid.pairs = 10, n.grid = min(round((5 + sqrt(NROW(x))) * 4), NROW(x)), ...)

## S4 method for signature 'matrix'
pdfCluster(x, graphtype, hmult, Q = "QJ", lambda = 0.1,
grid.pairs = 10, n.grid = min(round((5 + sqrt(NROW(x))) * 4), NROW(x)), ...)

## S4 method for signature 'data.frame'
pdfCluster(x, graphtype, hmult, Q = "QJ", lambda = 0.1,
grid.pairs = 10, n.grid = min(round((5 + sqrt(NROW(x))) * 4), NROW(x)), ...)

## S4 method for signature 'pdfCluster'
pdfCluster(x, graphtype, hmult, Q, lambda = 0.1,
grid.pairs, n.grid = min(round((5 + sqrt(NROW(x@x))) * 4), NROW(x@x)), ...)
```

Arguments

x	A vector, a matrix or a data frame of numeric data to be partitioned. Since density-based clustering is designed for continuous data only, if discrete data are provided, a warning message is displayed. Alternatively, x can be an object of pdfCluster-class itself, obtained when graphtype is set to "pairs". See Section Details below.
graphtype	Either "unidimensional", "delaunay" or "pairs", it defines the procedure used to build the graph associated with the data. If missing, a "delaunay" graph is built for data having dimension less than 7, otherwise a "pairs" graph is built. See details below. This argument has not to be set when x is of pdfCluster-class.
hmult	A shrink factor to be multiplied by the smoothing parameter h of function kepdf. If missing, it is taken to be 1 when data have dimension greater than 6, 0.75 otherwise.
Q	Optional arguments to be given when graphtype = "delaunay". See delaunayn in package geometry for further details. This argument has not to be set when graphtype = "pairs", when graphtype = "unidimensional" or when x is of pdfCluster-class.
lambda	Tolerance threshold to be used when graphtype = "pairs". An edge is set between two observations if the density function, evaluated along the segment linking them, does not exhibit any valley having a measure exceeding lambda. Its range is $[0, 1)$ but a value larger than 0.3 is not recommended; default value is set to 0.10. This argument has not to be set when graphtype = "delaunay" or graphtype = "unidimensional".
grid.pairs	When graphtype = "pairs", this arguments defines the length of the grid of points along the segment linking each pair of observations, on which the density is evaluated. Default is 10. This argument has not to be set when graphtype = "delaunay", when graphtype = "unidimensional" or when x is of pdfCluster-class.
n.grid	Defines the length of the grid on which evaluating the connected components of the density level sets. The default value is set to the minimum between the

number of data rows n and $\lfloor(5+\sqrt{(}n))4+0.5\rfloor$, an empirical rule of thumb which indicates that the length of the grid grows with the square root of the number of rows data.

... Further arguments to be passed to `kepdf` or to `pdfClassification`.

Details

Clusters are associated to the connected components of the level sets of the density underlying the data. Density estimation is performed by kernel methods and the connected regions are approximated by the connected components of a graph built on data. Three alternative procedures to build the graph are adopted:

Unidimensional procedure When data are univariate an edge is set between two observations when all the data points included in the segment between the two candidate observations belong to the same level set.

Delaunay triangulation An edge is set between two observations when they are contiguous in the Voronoi diagram; see Azzalini and Torelli (2007).

Pairs evaluation An edge is set between two observations when the density function, evaluated along the segment joining them, does not exhibit any valley having a relative amplitude greater than a tolerance threshold $0 \leq \lambda < 1$. Being a tolerance threshold, sensible values of λ are, in practice, included in $[0, 0.3]$; see Menardi and Azzalini (2013).

As the level set varies, the number of detected components gives rise to the tree of clusters, where each leave corresponds to a mode of the density function. Observations around these modes form a number of cluster cores, while the lower density observations are allocated according to a classification procedure; see also `pdfClassification`.

Value

An S4 object of `pdfCluster-class` with slots:

`call` The matched call.

`x` The matrix of data input. If a vector of data is provided as input, a one-column matrix is returned as output.

`pdf` An object of class `list` providing information about the density estimate. It includes:

- `kernel` character vector defining the kernel function used to estimate the density;
- `bwtype` character vector defining if a fixed or an adaptive kernel estimator has been used;
- `par` list of components defining the parameters used in density estimation;
- `estimate` vector of density estimates evaluated at the data points.

See `kepdf` for further details.

`nc` An object of class `list` defining details about the identification of the connected regions. It includes:

- `nc` number of connected sets for each density level set.

- p vector of level sets, giving the proportions of data with estimated density below a threshold.
- id group label of each point at different sections of the density estimate. Negative values of id mean that the estimated density is below the considered threshold.
- pq for each p gives the corresponding quantile q of the values of the density.

graph — An object of class list defining details about the graph built to find the connected sets of high density regions. Its length depends on the value of its first element:

- type either "unidimensional", "delaunay" or "pairs", defines the procedure used to set edges among the observations. In the last case only, the list includes also the following elements:
- comp.area a list containing the vector area and the matrix pairs.ord. The element i of vector area is the measure of the maximum valley in the density function linking the observations having row position as given in column i of pairs.ord.
- lambda tolerance threshold.

cluster.cores — A vector with the same length as NROW(x), defining the cluster cores membership. NA values correspond to low density, unlabeled data, to be classified in the second phase of the procedure by the intarnal call of pdfClassification.

tree — Cluster tree with leaves corresponding to the connected components associated to different sections of the density estimate. The object is of class dendrogram.

noc — Number of clusters.

stages — List with elements corresponding to the data allocation to groups at the different stages of the classification procedure. NA values correspond to unlabeled data.

clusters — Set to NULL if n.stages = 0, that is, if data belonging to the cluster cores only have been allocated. Otherwise it reports the final label groups. This component is obsolete. Use function groups, instead.

Methods

signature(x="data.frame") This method applies the pdfCluster procedure to objects of class data.frame.

signature(x="matrix") This method applies the pdfCluster procedure to objects of class matrix.

signature(x="numeric") This method applies the pdfCluster procedure to objects of class numeric.

signature(x="pdfCluster") This method applies to objects of pdfCluster-class when the graph has been built according to the "pairs" procedure. It allows to save time and computations if the user wants to compare results of cluster analysis for different values of the lambda parameter. See examples below.

Warning

It may happen that the variability of the estimated density is so high that not all jumps in the mode function can be detected by the selected grid scanning the density function. In that case, no

output is produced and a message is displayed. As this may be associated to the occurrence of some spurious connected components, which appear and disappear within the range between two subsequent values of the grid, a natural solution is to increase the value of `n.grid`. Alternatively either `lambda` or `hmult` may be increased to alleviate the chance of detecting spurious connected components.

Using `graphtype= 'delaunay'` when the dimensionality d of data is greater than 6 is highly time-consuming unless the number of rows n is fairly small, since the number of operations to run the Delaunay triangulation grows exponentially with d. Use `graphtype= "pairs"`, instead, whose computational complexity grows quadratically with the number of observations.

References

Azzalini, A., Menardi, G. (2014). Clustering via nonparametric density estimation: the R package pdfCluster. *Journal of Statistical Software*, 57(11), 1-26, URL http://www.jstatsoft.org/v57/i11/.

Azzalini A., Torelli N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*. 17, 71-80.

Menardi, G., Azzalini, A. (2014). An advancement in clustering via nonparametric density estimation. *Statistics and Computing*. DOI: 10.1007/s11222-013-9400-x.

See Also

kepdf, pdfCluster-class, pdfClassification.

Examples

```
##########
#example 1
############
# not run here for time reasons
#loading data
data(oliveoil)

#preparing data
olive1 <- 1 + oliveoil[, 3:10]
margin <- apply(data.matrix(olive1),1,sum)
olive1 <- olive1/margin
alr <- (-log( olive1[, -4]/olive1[, 4]))
#select the first 5 principal components
x <- princomp(alr, cor=TRUE)$scores[, 1:5]

#clustering
# not run here for time reasons
#cl <- pdfCluster(x, h = h.norm(x), hmult=0.75)
#summary(cl)
#plot(cl)

#comparing groups with original macro-area membership
#groups <- groups(cl)
#table(oliveoil$macro.area, groups)
```

```
#cluster cores
#table(groups(cl, stage = 0))

##########
#example 2
###########
# not run here for time reasons
# loading data
#data(wine)
#x <-wine[ ,-1]
#gr <- wine[ ,1]

# when data are high-dimensional, an adaptive kernel estimator is preferable
# building the Delaunay graph entails a too high computational effort
# use option "pairs" to build the graph
# it is the default option for dimension >6


# cl <- pdfCluster(x, graphtype="pairs", bwtype="adaptive")
# summary(cl)
# plot(cl)

#comparison with original groups
#table(groups(cl),gr)

# a better classification is obtained with larger value of lambda
# not necessary to run the whole procedure again
# a pdfCluster method applies on pdfCluster-class objects!

#cl1 <- pdfCluster(cl, lambda=0.25)
#table(gr, groups(cl1))
```

`pdfCluster-class` *Class "pdfCluster"*

Description

This class pertains to results of the application of function `pdfCluster`.

Objects from the Class

Objects can be created by calls of the form `new("pdfCluster", ...)` or as a result to a call to `pdfCluster`.

Slots

`call`: Object of class `"call"` representing the matched call;

`x`: Object of class `"matrix"` representing the clustered data points;

`pdf`: Object of class `"list"` reporting details about the kernel density estimate at data points `x`.

`nc`: Object of class `"list"` summarizing the result of the connected components search for different sections of the estimated density.

`graph`: An object of class `"list"` defining details about the graph built to find the connected sets of high density regions.

`cluster.cores`: Object of class `"ANY"` reporting the group labels of the data allocated to the cluster cores.

`tree`: Object of class `"ANY"`, namely class `dendrogram` if the procedure detects more than one group, `list` otherwise. It reports the cluster tree structure associated to the different connected components for different density levels.

`noc`: Object of class `"numeric"` giving the number of clusters.

`stages`: Object of class `"ANY"`, being `NULL` if the cluster cores only are detected, `"list"` when also the lower density data are allocated. The elements of the list correspond to the group labels at the different stages of the classification procedure. `NA` values correspond to unlabeled data.

`clusters`: Object of class `"ANY"` being `NULL` if the cluster cores only are detected, `"numeric"` when all the data are clustered. This slot is obsolete. Groups can be extracted by a call to function `groups`.

See `pdfCluster` for further details.

Methods

dbs `signature(x = "pdfCluster", clusters = "missing")`

Computes the density based Silhouette diagnostics of clustered data. See `dbs` for further details.

pdfCluster `signature(x="pdfCluster")`

Speeds up time for re-running the `pdfCluster` procedure with different values of `tau` when `graphtype = "pairs"`

plot `signature(x = "pdfCluster", y = "missing")`

Plots objects of `pdfCluster-class`. `plot-methods` are available for:

- the mode function: gives the number of connected components when the proportion of data points with density above a threshold varies. Set argument `which` to 1 to display this plot.
- the cluster tree: plot the hierarchical structure associated to the clusters detected by different sections of the density estimate. Set argument `which` to 2 to display this plot.
- the data points: scatterplot of data or of all the possible couples of coordinates reporting the label group. Set argument `which` to 3 to display this plot.
- the density-based Silhouette information: graphical diagnostics of the clustering. See `plot,dbs-method`. Set argument `which` to 4 to display this plot. Not available when `noc=1`.

See `plot,pdfCluster-method` for further details.

show `signature(object = "pdfCluster")`.

Prints the following elements:

- the matched Call;
- the type of kernel estimator;

- the type of graph built;
- the groups tree (if available);
- the cluster cores;
- the cluster labels at the different stages of the classification procedure;
- the final clustering.

summary `signature(object = "pdfCluster")`.

Provides a summary of `pdfCluster-class` objects by printing the following elements:

- the matched call to pdfCluster function
- the frequency table of the cluster cores;
- the frequency table of the final grouping;
- the tree of clusters.

See Also

`pdfCluster`, `plot,pdfCluster-method`, `show-methods`, `summary-methods`

Examples

```
showClass("pdfCluster")

data(wine)
x <-wine[ ,-1]
gr <- wine[ ,1]

# clustering
cl <- pdfCluster(x, graphtype="pairs", bwtype="adaptive")
summary(cl)
cl
plot(cl)
```

`plot,dbs-method` *Plot objects of class dbs*

Description

This function provides a graphical tool to display diagnostics of density-based cluster analysis by means of the density-based silhouette information.

Usage

```
## S4 method for signature 'dbs'
plot(x, y , xlab = "", ylab = "", col = NULL, lwd = 3, cex = 0.9,
    cex.axis = 0.5, main = NULL, labels = FALSE, ...)
```

Arguments

x	An object of dbs-class;
y	Not used; for compatibility with generic plot;
xlab	A title for the x axis;
ylab	A title for the y axis;
col	A specification for the plotting color. Default are colors in palette corresponding to the group labels;
lwd	A specification for the width of the bars in the plot;
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default;
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex;
main	An overall title for the plot;
labels	Logical. Should row index of data be added to the plot?
...	Further arguments to be passed to plot.

Details

After computing the density-based silhouette index by applying dbs-methods, data are partitioned into the clusters, sorted in a decreasing order with respect to their dbs value and displayed on a bar graph.

Methods

signature(x = "dbs", y = "missing") S4 method for plotting objects of dbs-class

See Also

dbs, dbs-class, silhouette.

Examples

```
#example 1: no groups in data
#random generation of group labels
set.seed(54321)
x <- rnorm(50)
groups <- sample(1:2, 50, replace=TRUE)
groups
dsil <- dbs(x=as.matrix(x), clusters=groups)
dsil
summary(dsil)
plot(dsil, labels=TRUE, lwd=6)

#example 2: wines data
# load data
data(wine)
```

```
gr <- wine[,1]

# select a subset of variables
x <- wine[, c(2,5,8)]

#clustering
cl <- pdfCluster(x)

dsil <- dbs(cl)
plot(dsil)
```

---

plot,kepdf-method *Plot objects of class kepdf*

Description

Functions and methods for plotting objects of kepdf-class.

Usage

```
## S4 method for signature 'kepdf'
plot(x, y, eval.points = NULL, n.grid = NULL,
data = NULL, add = FALSE, main = NULL, xlab = NULL, ylab = NULL,
zlab = NULL, col = NULL, col.data=2, type="l", props = c(75,50,25),
method="contour", ticktype = "detailed", indcol = NULL,
text.diag.panel = NULL, gap = 0.5, ...)
```

Arguments

x	An object of kepdf-class;
y	Not used; for compatibility with generic plot;
eval.points	A matrix of data points at which the density to be plotted has to be evaluated; the number of columns must correspond to the dimension of the sample data used to estimate the density. If not provided, density is evaluated on a grid defined on the range of sample data.
n.grid	A vector with length set to the number of column of sample data, defining the length of the grid on which the density to be plotted is evaluated; this argument is ignored when eval.points is not NULL.
data	Data to be optionally superimposed to the density plot.
add	Logical. If TRUE, add to a current plot.
main	An overall title for the plot.
xlab	A title for the x axis.
ylab	A title for the y axis.
zlab	A title for the z axis.
col	A specification for the plotting color.

col.data	A specification for the color of data. Ignored if data is NULL.
type	What type of plot should be drawn. This argument applies when kernel density estimate is performed on unidimensional data only. Default value is "l".
props	A vector defining the fraction of the data to be included within each density level. This argument applies when kernel density estimate is performed on multidimensional data only.
method	One of c("contour", "image", "perspective"). To be used when two or higher dimensional data have been used to estimate the density.
ticktype	Character: "simple" draws just an arrow parallel to the axis to indicate direction of increase; "detailed" draws normal ticks; to be used if method="perspective" only.
indcol	Vector of the column positions to be plotted, when densities are estimated on higher than 2-dimensional data.
text.diag.panel	Text to be displayed on the diagonal panels when plotting densities estimated on higher than 2-dimensional data.
gap	Distance between subplots, when plotting densities estimated of 2-dimensional data or higher-dimensional data.
...	Further arguments to be passed to plot, image, contour, persp.

Details

When density estimation is based on two or higher dimensional data, these functions make use of functions contour, image and persp.

For densities estimated on higher than 2-d data, the pairwise marginal estimated densities are plotted for all possible pairs of coordinates or a chosen selection of them.

Value

A list containing the following elements:

eval.points	data points at which the plotted density has been evaluated
estimate	the estimated density at eval.points

Methods

signature(x = "kepdf", y = "missing") S4 method for plotting objects of kepdf-class.

See Also

kepdf-class, plot, contour, image, plot-methods,persp

Examples

```
#1-d example
set.seed(123)
x1 <- rnorm(50)

#normal optimal bandwidth
pdf1a <- kepdf(x1)
#shrink the smoothing parameter
pdf1b <- kepdf(x1, h=0.5*h.norm(x1))

plot(pdf1a, n.grid=50, data=x1, xlab="x1", ylim=c(0, max(c(pdf1a@estimate,
   pdf1b@estimate))))
plot(pdf1b, n.grid=50, lty=2, add=TRUE)

#2-d example
set.seed(123)
x2 <- cbind(rnorm(50),rnorm(50))

pdf2 <- kepdf(x2)

plot(pdf2, n.grid=c(50,50), data=x2)
plot(pdf2, n.grid=c(50,50), method="image")
plot(pdf2, n.grid=c(50,50), method="perspective", phi=30, theta=30)

#3-d example
set.seed(123)
x3 <- cbind(rnorm(50), rnorm(50), rnorm(50))

pdf3 <- kepdf(x3)

plot(pdf3, n.grid=c(50,50,50))
plot(pdf3, n.grid=c(50,50,50), method="image", col = terrain.colors(30))
plot(pdf3, n.grid=c(50,50,50), method="perspective", phi=30, theta=30)
```

`plot,pdfCluster-method`

Plot objects of class pdfCluster

Description

Functions and methods for plotting objects of `pdfCluster-class`.

Usage

```
## S4 method for signature 'pdfCluster'
plot(x, y, which = 1:4, stage = Inf, pch = NULL, col = NULL, ...)
```

Arguments

x	An object of pdfCluster-class;
y	Not used; for compatibility with generic plot;
which	To be used to select the type of plot: • when which = 1 plots the mode function, corresponding to the number of modes for different proportions of data points with density above a threshold. • when which = 2 plots the cluster tree associated to different sections of the density estimate. • when which = 3 displays the scatterplot of data or of all the possible pairs of coordinates reporting the label group. • when which = 4 the plot.dbs is displayed. Multiple choices are possible.
stage	Plots the data points at the indicated stage of the classification procedure. Unallocated data are indicated by 0s. This argument applies if which=3 only.
pch	Either an integer specifying a symbol or a single character to be used in plotting points. If a vector of the same length as the number of groups is given, different symbols or characters are used for different groups. The default value denotes points as their group label. This argument applies if which=3 only.
col	Colors to be used in plotting points. If a vector of the same length as the number of groups is given, different colors or characters are used for different groups. The default value use colors in palette corresponding to the the group labels of the data. This argument applies if which=3 only.
...	Further arguments to be passed to plot-methods.

Methods

signature(x = "pdfCluster", y = "missing") S4 method for plotting objects of pdfCluster-class

References

Azzalini A., Torelli N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*. vol. 17, pp. 71-80.

See Also

pdfCluster-class, plot, plot-methods.

Examples

```
data(wine)
gr <- wine[,1]

# select a subset of variables
x <- wine[, c(2,5,8)]
```

```
#clustering
cl <- pdfCluster(x)
plot(cl, which=3, stage=2)

table(cl@clusters, gr)
#set "B" for Barolo, "G" for Grignolino, "A" for Barbera
plot(cl,  pch=c("B", "G", "A"), col=c(3,4,5))
```

---

plot-methods *Methods for function plot*

Description

Methods for functions `plot` aimed at graphically displaying the S4 classes included in the `pdfCluster-package`.

Methods

`signature(x = "kepdf", y = "ANY")` S4 method for plotting objects of `kepdf-class`. See `plot,kepdf-method` for further details.

`signature(x = "dbs", y = "missing")` S4 method for plotting objects of `kepdf-class`. See `plot,dbs-method` for further details.

`signature(x = "pdfCluster", y = "missing")` S4 method for plotting objects of `pdfCluster-class`. See `plot,pdfCluster-method` for further details.

See Also

`plot,dbs-method`, `plot,kepdf-method`, `plot,pdfCluster-method`

---

show-methods *Methods for Function show*

Description

Methods for function `show` aimed at showing the S4 classes included in the `pdfCluster-package`.

Methods

`signature(object = "kepdf")` S4 method for showing objects of `kepdf-class`.

`signature(object = "dbs")` S4 method for showing objects of `dbs-class`.

`signature(object = "pdfCluster")` S4 method for showing objects of `pdfCluster-class`.

See Also

`dbs-class`, `kepdf-class`, `pdfCluster-class`

summary-methods *Methods for Function summary*

Description

Methods for function summary aimed at summarizing the S4 classes included in the pdfCluster-package.

Methods

signature(object = "dbs") S4 method for summarizing objects of dbs-class.

signature(object = "kepdf") S4 method for summarizing objects of kepdf-class.

signature(object = "pdfCluster") S4 method for summarizing objects of pdfCluster-class.

See Also

dbs-class, kepdf-class, pdfCluster-class.

wine *Wine data*

Description

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wine: Barolo, Grignolino, Barbera. The data set is used to evaluate the pdfCluster ability of understanding the type of wine, given the chemical measurement.

Usage

```
data(wine)
```

Format

This data frame contains 178 rows, each corresponding to a different cultivar of wine produced in Piedmont (Italy), and 14 columns. The first column is the type of wine, a factor variable with the following levels: Barolo, Grignolino, Barbera. The variables measured on the three types of wines are the following: Alcohol, Malic acid, Ash, Alcalinity, Magnesium, Phenols, Flavanoids, Nonflavanoids, Proanthocyanins, Color intensity, Hue, OD280.OD315Dilution, Proline. All variables but the label class are continuous.

Details

The original data set comprises 27 variables. Here a subset of 14 variables only has been included.

Source

Forina, M., Lanteri, S. Armanino, C., Casolino, C., Casale, M., Oliveri, P. (2008). V-PARVUS. *An Extendible Pachage of programs for esplorative data analysis, classification and regression analysis*. Dip. Chimica e Tecnologie Farmaceutiche ed Alimentari, Università di Genova.

Index