Package 'FCPS'

March 13, 2020

Type Package

Title Fundamental Clustering Problems Suite

Version 1.1.0

Date 2020-03-13

Maintainer Michael Thrun <m.thrun@gmx.net>

Description Many conventional Clustering Algorithms are provided in this package with consistent input and output, which enables the user to tryout algorithms swiftly. Additionally, 26 statistical approaches for the estimation of the number of clusters as well as the the mirrored density plot (MD-plot) of clusterability are provided. Moreover, the fundamental clustering problems suite (FCPS) offers a variety of clustering challenges any algorithm should handle when facing real world data. Nine of the here presented artificial datasets were named FCPS in Ultsch, A.: ``Clustering with SOM: U*C", In Workshop on Self-Organizing Maps, 2005.

Imports mclust, ggplot2, DataVisualizations

Suggests kernlab, cclust, vegan, dbscan, kohonen, MCL, ADPclust, cluster, DatabionicSwarm, orclus, subspace, flexclust, ABCanalysis, apcluster, pracma,EMCluster, pdfCluster, parallelDist, plotly, ProjectionBasedClustering, GeneralizedUmatrix, mstknnclust, densityClust, parallel, energy, R.utils, tclust, Spectrum, genie, protoclust, fastcluster, clusterability, signal, reshape2

Depends R (>= 3.5.0)

License GPL-3

LazyData TRUE

LazyLoad yes

NeedsCompilation no

Author Michael Thrun [aut, cre, cph], Peter Nahrgang [ctr, ctb], Alfred Ultsch [dtc, ctb]

Repository CRAN

Date/Publication 2020-03-13 12:50:02 UTC

R topics documented:

`FCPS-package` *FCPC Data Sets with Common Clustering Algorithms*

Description

The 'Fundamental Clustering Problems Suite' (FCPS) originally offered a variety of clustering problems any algorithm shall be able to handle when facing real world data. FCPS served as an elementary benchmark for clustering algorithms.

The FCPS package extends datasets and provides a standardized and easy access to many clustering algorithms.

FCPS datasets consists of data sets with known a priori classifications that are to be reproduced by the algorithm. All data sets are intentionally created to be simple and might be visualized in two or three dimensions. Each data sets represents a certain problem that is solved by known clustering algorithms with varying success. This is done in order to reveal benefits and shortcomings of algorithms in question. Standard clustering methods, e.g. single-linkage, ward and k-means, are not able to solve all FCPS problems satisfactorily.

"Lsun3D and each of the nine artificial data sets of "Fundamental Clustering Problems Suite" (FCPS) were defined separately for a specific clustering problem as cited (in [Thrun/Ultsch, 2020]), but nine of the here presented artificial datasets were named FCPS in [Ultsch, 2005]. The original sample size defined in the respective first publication mentioning the data was used in [Thrun/Ultsch, 2020], but using the R function "GenerateFundamentalClusteringProblem" (...) any sample size can be drawn for all artificial data sets. " [Thrun/Ultsch, 2020]

Author(s)

Authors@R: c(person("Michael", "Thrun", email= "m.thrun@gmx.net",role=c("aut","cre","cph")),person("Peter", "Nahrgang",role=c("ctr","ctb")),person("Alfred", "Ultsch",role=c("dtc","ctb")))

Maintainer: Michael Thrun <m.thrun@gmx.net>

References

[Thrun, 2018] Thrun, M. C.: Projection Based Clustering through Self-Organization and Swarm Intelligence, doctoral dissertation 2017, Springer, ISBN: 978-3-658-20539-3, Heidelberg, 2018.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

[Ultsch, 2005] Ultsch, A.: Clustering with SOM: U*C, In Proc. Workshop on Self-Organizing Maps, pp. 75-82, Paris, France, 2005.

`ADPclustering` *(Adaptive) Density Peak Clustering algorithm using Automatic Parameter Selection*

Description

The algorithm was introduced in [Rodriguez/Laio, 2014] and here implemented by [Wang/Xu, 2017]. The algorithm is adaptive in the sense that only `ClusterNo` has to be set instead of the paramters of [Rodriguez/Laio, 2014] implemented in `ADPclustering`.

Usage

```
ADPclustering(Data,ClusterNo=NULL,PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	Optional, Either: A number k which defines k different Clusters to be build by the algorithm, or a range of `ClusterNo` to let the algorithm choose from.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

The ADP algorithm decides the k number of clusters. This contrary to the other version of the algorithm of another package which can be called with `DensityPeakClustering`.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Rodriguez/Laio, 2014] Rodriguez, A., & Laio, A.: Clustering by fast search and find of density peaks, Science, Vol. 344(6191), pp. 1492-1496. 2014.

[Wang/Xu, 2017] Wang, X.-F., & Xu, Y.: Fast clustering using adaptive density peak detection, Statistical methods in medical research, Vol. 26(6), pp. 2800-2811. 2017.

See Also

DensityPeakClustering

adpclust

Examples

```
data('Hepta')
out=ADPclustering(Hepta$Data,PlotIt=FALSE)
```

AgglomerativeNestingClustering

AGNES clustering

Description

agglomerative hierarchical clustering (AGNES)

Usage

```
AgglomerativeNestingClustering(DataOrDistances, ClusterNo,

PlotIt = FALSE, Standardization = TRUE, Data, ...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm. if ClusterNo=0, the dendrogram is generated instead of a clustering to estimate the numbers of clusters.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls

`Standardization`
: `DataOrDistances` is standardized before calculating the dissimilarities. Measurements are standardized for each variable (column), by subtracting the variable's mean value and dividing by the variable's mean absolute deviation.If `DataOrDistances` is already a distance matrix, then this argument will be ignored.

`Data`
: [1:n,1:d] data matrix in the case that `DataOrDistances` is missing and partial matching does not work.

`...`
: Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

`Cls`
: [1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.

`Object`
: Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

Kaufman, L. and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.

Anja Struyf, Mia Hubert and Peter J. Rousseeuw (1996) Clustering in an Object-Oriented Environment. Journal of Statistical Software 1. 10.18637/jss.v001.i04

Struyf, A., Hubert, M. and Rousseeuw, P.J. (1997). Integrating Robust Clustering Techniques in S-PLUS, Computational Statistics and Data Analysis, 26, 17–37.

Lance, G.N., and W.T. Williams (1966). A General Theory of Classifactory Sorting Strategies, I. Hierarchical Systems. Computer J. 9, 373–380.

Belbin, L., Faith, D.P. and Milligan, G.W. (1992). A Comparison of Two Approaches to Beta-Flexible Clustering. Multivariate Behavioral Research, 27, 417–433.

See Also

agnes

Examples

```
data('Hepta')
out=AgglomerativeNestingClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

print(out$agnesObject)
plot(out$agnesObject)
```

`APclustering` *Affinity Propagation clustering*

Description

Affinity Propagation clustering published by [Frey/Dueck, 2007] and implemented by [Bodenhofer et al., 2011].

Usage

```
APclustering(DataOrDistances,

InputPreference=NA,ExemplarPreferences=NA,

DistanceMethod="euclidean",

Seed=7568,PlotIt=FALSE,Data,...)
```

Arguments

`DataOrDistances`	[1:n,1:d] with: if d=n and symmetric then distance matrix assumed, otherwise: [1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. In the later case the Euclidean distances will be calculated.
`InputPreference`	default parameter set, see **apcluster**
`ExemplarPreferences`	default parameter set, see **apcluster**
`DistanceMethod`	DistanceMethod as in `dist` for `similarities`.
`Seed`	set as integervalue to have reproducible results, see **apcluster**
`PlotIt`	default: FALSE, If TRUE and dataset of [1:n,1:d] dimensions then a plot of the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls` will be generated.
`Data`	[1:n,1:d] data matrix in the case that `DataOrDistances` is missing and partial matching does not work.
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

Distancematrix D is converted to similaritiy matrix S with S=-(D^2).

If Data matrix is used, then euclidean similarities are calculated by `similarities` and a specifed distance method.

The AP algorithm decides the k number of clusters.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Frey/Dueck, 2007] Frey, B. J., & Dueck, D.: Clustering by passing messages between data points, Science, Vol. 315(5814), pp. 972-976, <doi:10.1126/science.1136800>, 2007.

[Bodenhofer et al., 2011] Bodenhofer, U., Kothmeier, A., & Hochreiter, S.: APCluster: an R package for affinity propagation clustering, Bioinformatics, Vol. 27(17), pp, 2463-2464, 2011.

further Details in http://www.bioinf.jku.at/software/apcluster

See Also

apcluster

Examples

```
data('Hepta')
res=APclustering(Hepta$Data)

library(DataVisualizations)
DataVisualizations::Plot3D(Hepta$Data,res$Cls)
```

`Atom` *Atom of [Ultsch, 2004].*

Description

Two nested spheres with different variances that are not linear not separable. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Atom")
```

Details

Size 800, Dimensions 3, stored in `Atom$Data`

Classes 2, stored in `Atom$Cls`

References

[Ultsch, 2004] Ultsch, A.: Strategies for an artificial life system to cluster high dimensional data, Abstracting and Synthesizing the Principles of Living Systems, GWAL-6, U. Brggemann, H. Schaub, and F. Detje, Eds, pp. 128-137. 2004.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Atom)
str(Atom)
```

Chainlink *Chainlink of [Ultsch et al., 1994; Ultsch, 1995].*

Description

Two chains of rings. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Chainlink")
```

Details

Size 1000, Dimensions 3, stored in `Chainlink$Data`

Classes 2, stored in `Chainlink$Cls`

References

[Ultsch et al., 1994] Ultsch, A., Guimaraes, G., Korus, D., & Li, H.: Knowledge extraction from artificial neural networks and applications, Parallele Datenverarbeitung mit dem Transputer, (pp. 148-162), Springer, 1994.

[Ultsch, 1995] Ultsch, A.: Self organizing neural networks perform different from statistical k-means clustering, Proc. Society for Information and Classification (GFKL), Vol. 1995, Basel 8th-10th March 1995.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Chainlink)
str(Chainlink)
```

`ClusterabilityMDplot` *Clusterability MDplot*

Description

Clusterability Mirrored-Density Plot. Clusterability aims to quantify the degree of cluster structures [Adolfsson et al., 2019]. A dataset has a high probabilty to posses cluster structures, if the first component of the PCA projection is multimodal [Adolfsson et al., 2019]. As the dip test ist less exact than the MDplot [Thrun et al., 2019] , pvalues above 0.05 can be given for MDplots which are clearly multimodal.

An alternative investigation of clusterability can be performed by inspecting the topographic map of the Generalized U-Matrix for a specfic projection method using the **ProjectionBasesdClustering** and **GeneralizedUmatrix** packages on CRAN, see [Thrun/Ultsch, 2020] for details.

Usage

```
ClusterabilityMDplot(Data,Method)
```

Arguments

`Data`	einer one datasets [1:n,1:d] of n cases and d features or multiple data sets in a list
`Method`	"none" performs no dimension reduction. "pca" uses the scores from the first principal component. "distance" computes pairwise distances (using distance_metric as the metric).

Details

Uses the method of [Adolfsson et al., 2019] sepecified as pca plus dip-test (PCA dip).

If list is named, than the names of the list will be used and the MDplots will be re-ordered according to multimodality in the plot, otherwise only the pvalues of [Adolfsson et al., 2019] will be the names and the ordering of the MDplots is the same as the of the list.

Beware, as shown below, this test fails for alomost touching clusters of Tetra and is difficult to intepret on WingNut but with overlayed with a roubustly estimated unimodal Gaussian distribution it can be interpreted as multimodal). However, it does not fail for chaining data contrary to the claim in [Adolfsson et al., 2019].

Value

ggplot2 plotter handle

Note

Based on work currently under review, the author of this function disagrees with [Adolfsson et al., 2019] as to the preference which clusterablity method should be used and approach "distance" is not preferable for density-based cluster structures.

Author(s)

Michael Thrun

References

[Adolfsson et al., 2019] Adolfsson, A., Ackerman, M., & Brownstein, N. C.: To cluster, or not to cluster: An analysis of clusterability methods, Pattern Recognition, Vol. 88, pp. 13-26. 2019.

[Thrun et al., 2019] Thrun, M. C., Gehlert, T., & Ultsch, A.: Analyzing the Fine Structure of Distributions, preprint available at arXiv.org, Vol. under review, pp. arXiv:1908.06081. doi arXiv:1908.06081, 2019.

[Thrun/Ultsch, 2020] Thrun, M. C., and Ultsch, A.: Swarm Intelligence for Self-Organized Clustering, Artificial Intelligence, in press, https://doi.org/10.1016/j.artint.2020.103237, 2020.

Examples

```
##one dataset
data(Hepta)

ClusterabilityMDplot(Hepta$Data)

##multiple datasets
data(Atom)
data(Chainlink)
data(Lsun3D)
data(GolfBall)
data(EngyTime)
data(Target)
data(Tetra)
data(WingNut)
data(TwoDiamonds)

DataV = list(
  Atom = Atom$Data,
  Chainlink = Chainlink$Data,
  Hepta = Hepta$Data,
  Lsun3D = Lsun3D$Data,
  GolfBall = GolfBall$Data,
  EngyTime = EngyTime$Data,
  Target = Target$Data,
  Tetra = Tetra$Data,
  WingNut = WingNut$Data,
  TwoDiamonds = TwoDiamonds$Data
)
```

```
ClusterabilityMDplot(DataV)
```

`ClusterDistances` *ClusterDistances*

Description

Computes intra-cluster distances which are the distance in-between each cluster.

Usage

```
ClusterDistances(FullDistanceMatrix, Cls,

Names, PlotIt = FALSE)
```

Arguments

`FullDistanceMatrix`	[1:n,1:n] symmetric distance matrix
`Cls`	[1:n] numerical vector of k classes
`Names`	Optional [1:k] character vector naming k classes
`PlotIt`	Optional, Plots if TRUE

Details

Cluster distances are given back as a matrix, one column per cluster and the vector of the full distance matrix without the diagonal elements and the upper half of the symmetric matrix.

Value

matrix [1:m,1:(k+1)] of k clusters, each columns consists of the distances in a cluster, filled up with NaN at the end to be of the same lenght as the complete distance matrix.

Note

PlotIt is under development

Author(s)

Michael Thrun

References

[Thrun, 2018] Thrun, M.C., Projection Based Clustering through Self-Organization and Swarm Intelligence. 2018, Heidelberg: Springer.

See Also

MDplot

Examples

```
##ToDo
```

ClusteringAccuracy *ClusteringAccuracy*

Description

ClusteringAccuracy

Usage

```
ClusteringAccuracy(PriorCls,CurrentCls,K=9)
```

Arguments

PriorCls	
CurrentCls	clustering result
K	Maximal number of classes for computation.

Details

Here, accuracy is defined as the normalized sum over all true positive labeled data points of a clustering algorithm. The best of all permutation of labels with the highest accuracy is selected in every trial because algorithms arbitrarily define the labels.

Value

Accuracy Between zero and one

Author(s)

Michael Thrun

References

Michael C. Thrun, Felix Pape, Alfred Ultsch: Benchmarking Cluster Analysis Methods in the Case of Distance and Density-based Structures Defined by a Prior Classification Using PDE-Optimized Violin Plots, ECDA, Potsdam, 2018

Examples

```
data(Hepta)

InputDistances=as.matrix(dist(Hepta$Data))
projection=Pswarm(InputDistances)
visualization=GeneratePswarmVisualization(Data = Hepta$Data,

projection$ProjectedPoints,projection$LC)
Cls=DBSclustering(k=7, Hepta$Data, visualization$Bestmatches,

visualization$LC,PlotIt=FALSE)
ClusteringAccuracy(Hepta$Cls,Cls,K=9)
```

Clusternumbers *Estimates Number of Clusters using up to 26 Indicators*

Description

Calculation of up to 26 indicators and the recommendations based on them for the number of clusters in data sets. For a given dataset and clusterings for this dataset, key indicators mentioned in details are calculated and based on this a recommendation regarding the number of classes is given for each indicator.

An alternative estimation of the cluster number can be done by counting the valleys of the topographic map of the generalized U-Matrix for a specfic projection method using the **ProjectionBasesdClustering** and **GeneralizedUmatrix** packages on CRAN, see [Thrun/Ultsch, 2020] for details.

Usage

```
Clusternumbers(Data, ClsMatrix = NULL, max.nc,

index = "all", min.nc = 2,

Silent = TRUE, method = NULL, PlotIt=TRUE)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
ClsMatrix	[1:n,1:(max.nc)] Clustering of the number of classes to be checked as a matrix with one cluster per column (see also details (2) and (3)), must be specified if method = NULL
max.nc	highest number of classes to be checked
method	Cluster procedure, with which the clusterings are created (see details (4) for possible methods), must be specified if ClsMatrix = NULL
	Optional:

`index`	String or vector of strings with the indicators to be calculated (see details (1)), default = "all
`min.nc`	lowest number of classes to be checked, default = 2
`Silent`	if TRUE status messages are output, default = FALSE
`PlotIt`	if TRUE plots fanplot with proposed cluster numbers

Details

(1)

The following 26 indicators can be calculated: "ball", "beale", "calinski", "ccc", "cindex", "db", "duda", "dunn", "frey", "friedman", "hartigan", "kl", "marriot", "mcclain", "pseudot2", "ptbiserial", "ratkowsky", "rubin", "scott", "sdbw", "sdindex", "silhouette", "ssi", "tracew", "trcovw", "xuindex".

These can be specified individually or as a vector via the parameter index. If you enter 'all', all key figures are calculated.

(2)

The codes kl, duda, pseudot2, beale, frey and mcclain require a clustering for max.nc+1 classes. If these key figures are to be calculated, this clustering must be specified in cls.

(3)

The code kl requires a clustering for min.nc-1 classes. If this key figure is to be calculated, this clustering must also be specified in cls. For the case min.nc = 2 no clustering for 1 has to be given.

(4)

The following methods can be used to create clusterings:

"ward.D", "single", "complete", "average", "mcquitty" "median," "centroid," "ward.D2," "kmeans," "DBSclustering,"

(5)

The indicators duda, pseudot2, beale and frey are only intended for use in hierarchical cluster procedures.

Value

`Indicators`	a table of the calculated indicators except Duda, Pseudot2 and Beale
`ClusterNo`	the recommended number of clusters for each calculated indicator
`ClsMatrix`	[1:n,min.nc:(max.nc)] Output of the clusterings used for the calculation
`HierarchicalIndicators`	Either NULL or the values for the indicators Duda, Pseudot2 and Beale in case of hierarchical cluster procedures, if calculated

Note

Code of "calinski", "cindex", "db", "hartigan", "ratkowsky", "scott", "marriot", "ball", "trcovw", "tracew", "friedman", "rubin", "ssi" of package cclust ist adapted for the purpose of this function.

Author(s)

Peter Nahrgang

References

Charrad, Malika, et al. "Package 'NbClust', J. Stat. Soft Vol. 61, pp. 1-36, 2014.

Dimtriadou, E. "cclust: Convex Clustering Methods and Clustering Indexes." R package version 0.6-16, URL https://CRAN.R-project.org/package=cclust, 2009.

[Thrun/Ultsch, 2020] Thrun, M. C., and Ultsch, A.: Swarm Intelligence for Self-Organized Clustering, Artificial Intelligence, in press, https://doi.org/10.1016/j.artint.2020.103237, 2020.

Examples

```
# Reading the iris dataset from the standard R-Package datasets
data <- as.matrix(iris[,1:4])

# Creating the clusterings for the data set
#(here with method complete) for the number of classes 2 to 8
hc <- hclust(dist(data), method = "complete")
clsm <- matrix(data = 0, nrow = dim(data)[1],  ncol = 7)
for (i in 2:8) {
  clsm[,i-1] <- cutree(hc,i)
}

# Calculation of all indicators and recommendations for the number of classes
indicatorsList=Clusternumbers(Data = data, ClsMatrix = clsm, max.nc = 7)

# Alternatively, the same calculation as above can be executed with the following call
Clusternumbers(Data = data, max.nc = 7, method = "complete")
#In this variant, the function clusterumbers also takes over the clustering
```

DBscan *DBscan*

Description

DBscan clustering

Usage

```
DBscan(Data,Radius,minPts,

PlotIt=FALSE,UpperLimitRadius,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
Radius	eps [Ester et al., 1996, p. 227] neighborhood in the R-ball graph/unit disk graph), size of the epsilon neighborhood. If missing, automatic estimation is done using insights of [Ultsch, 2005].

`minPts`	number of minimum points in the eps region (for core points). In principle minimum number of points in the unit disk, if the unit disk is within the cluster (core) [Ester et al., 1996, p. 228]. Default is 2.5 percent of points.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`UpperLimitRadius`	Limit for radius search, experimental
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

`Cls`	[1:n] numerical vector defining the clustering; this classification is the main output of the algorithm. Points which cannot be assigned to a cluster will be reported as members of the noise cluster with 0.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, Proc. Kdd, Vol. 96, pp. 226-231, 1996.

[Ultsch, 2005] Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery, In Baier, D. & Werrnecke, K. D. (Eds.), Innovations in classification, data science, and information systems, (Vol. 27, pp. 91-100), Berlin, Germany, Springer, 2005.

Examples

```
data('Hepta')

out=DBscan(Hepta$Data,PlotIt=FALSE)


#search for right parameter setting by grid search
data("WingNut")
Data = WingNut$Data
DBSGrid <- expand.grid(
  Radius = seq(from = 0.01, to = 0.3, by = 0.02),
  minPTs = seq(from = 1, to = 50, by = 2)
)
BestAcc = c()
for (i in seq_len(nrow(DBSGrid))) {
  print(i)
  parameters <- DBSGrid[i,]
  Cls9 = DBscan(
```

```
    Data,
    minPts = parameters$minPTs,

    Radius = parameters$Radius,
    PlotIt = F,

    UpperLimitRadius = parameters$Radius
  )$Cls
  if (length(unique(Cls9)) < 5)
    BestAcc[i] = ClusteringAccuracy(WingNut$Cls,

                                    Cls9) * 100
  else
    BestAcc[i] = 50
}
max(BestAcc)
which.max(BestAcc)
parameters <- DBSGrid[13,]

Cls9 = DBscan(
  Data,
  minPts = parameters$minPTs,
  Radius = parameters$Radius,
  PlotIt = F,
  UpperLimitRadius = parameters$Radius,
  PlotIt = TRUE
)$Cls
```

DBSclusteringAndVisualization

Databionic Swarm (DBS) Clustering and Visualization

Description

Swarm-based clustering by exploting self-organization, emergence, swarm intelligence and game theory.

Usage

```
DatabionicSwarmClustering(DataOrDistances, ClusterNo = 0,

StructureType = TRUE, DistancesMethod = NULL,

PlotTree = FALSE, PlotMap = FALSE,Data)
```

Arguments

DataOrDistances

Either nonsymmetric [1:n,1:d] datamatrix of n cases and d features or

	symmetric [1:n,1:n] distance matrix
`ClusterNo`	Number of Clusters, if zero a the topographic map is ploted. Number of valleys equals number of clusters.
`StructureType`	Either TRUE or FALSE, has to be tested against the visualization. if colored points of clusters a divided by mountain ranges, parameter is incorrect.
`DistancesMethod`	Optional, if data matrix given, annon Euclidean distance can be selected
`PlotTree`	Optional, if TRUE: dendrogram is plotted
`PlotMap`	Optional, if TRUE: topographic map is plotted.
`Data`	[1:n,1:d] data matrix in the case that `DataOrDistances` is missing and partial matching does not work.

Details

This function does not enable the user first to project the data and then to test the Boolean parameter defining the type of structure contrary to the **DatabionicSwarm** which is an inappropriate approach in case of exploratory data analysis.

Instead, this function is implemented for the purpose of automatic benchmarking because in such a case nobody will investigate many trials with one visualization per trial.

If one would like to perform a clustering exploratively (in the sense that a prior clustering is not given of evaluation purposes), then please use the **DatabionicSwarm** package directly and read the vignette there. Databionic swarm is like k-means an stochastic algorithm meaning that the clustering and visualization may change between trials.

Value

List of

`Cls`	[1:n] numerical vector of k clusters
`Object`	List of further output of DBS

Note

Current implementation is not efficient enough to cluster more than N=4000 cases as in that case in takes longer than a day for a result.

Author(s)

Michael Thrun

References

Thrun, M. C., & Ultsch, A.: Swarm Intelligence for Self-Organized Clustering, Journal of Artificial Intelligence, under minor revison, 2019.

See Also

`Pswarm`, `DBSclustering`,`GeneratePswarmVisualization`

Examples

```
#Generate random but small non-structured data set
data = cbind(
  sample(1:100, 300, replace = T),
  sample(1:100, 300, replace = T),
  sample(1:100, 300, replace = T)
)
#make sure there are no structures
# (sample size is small and still could generate structures randomly)
Data = RobustNormalization(data, Centered = TRUE)
#DataVisualizations::Plot3D(Data)

#No structres are visible
#topographic map looks like "egg carton"
# with every point in its own valley
Cls = DatabionicSwarmClustering(Data, 0, PlotMap = T)


#distance based cluster structures
#7 valleys are visible, thus ClusterNo=7

data(Hepta)
#DataVisualizations::Plot3D(Hepta$Data)

Cls = DatabionicSwarmClustering(Hepta$Data, 0, PlotMap = T)


#entagled, complex, and non-linear seperable structures

data(Chainlink)
#DataVisualizations::Plot3D(Chainlink$Data)

#2 valleys are visible, thus ClusterNo=2
Cls = DatabionicSwarmClustering(Chainlink$Data, 0, PlotMap = T)

#Trying of only parameter StructureType
#reveals that clustering is appropriate
# if StructureType=FALSE
Cls = DatabionicSwarmClustering(Chainlink$Data,
                                2,
                                StructureType = FALSE,
                                PlotMap = T)

#Here clusters (colored points)
#are not seperated by valleys
Cls = DatabionicSwarmClustering(Chainlink$Data,
                                2,
                                StructureType = TRUE,
                                PlotMap = T)
```

DensityPeakClustering *Density Peak Clustering algorithm using the Decision Graph*

Description

Density Peaks Clustering of [Rodriguez/Laio, 2014] is here implemented by [Pedersen et al., 2017] with estimation of [Wang et al, 2015] meaning its non adaptive in the sense of ADPclustering.

Usage

```
DensityPeakClustering(DataOrDistances, Rho,Delta,Dc,Knn=7,

method = "euclidean", PlotIt = FALSE, Data, ...)
```

Arguments

DataOrDistances	Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] not symmetric data matrix of n cased and d variable
Rho	local density of a point, See [Rodriguez/Laio, 2014] for explanation
Delta	minimum distance between a point and any other point, See [Rodriguez/Laio, 2014] for explanation
Dc	Optional, Cutoff distance, will either be estimated by [Pedersen et al., 2017] or [Wang et al, 2015] (see example below)
Knn	Optional k nearest neighbors
method	Optional distance method of data, default is euclid, see parDist for details
PlotIt	Optional TRUE: Plots 2d or 3d result with clustering
Data	[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial matching does not work.
...	Optional, further arguments for densityClust

Details

The densityClust algorithm does not decide the k number of clusters, this has to be done by the parameter setting. This contrary to the other version of the algorithm of another package which can be called with ADPclustering.

The plot shows the density peaks (Cluster centers). Set Rho and Delta as boundaries below the number of relevant cluster centers for your problem. (see example below).

Value

If Rho and Delta are set:

list of

Cls	Clustering as a numeric vector of k clusters

`Object` output of [Pedersen et al., 2017] algorithm

If Rho and Delta are missing:

`p` object of `plot_ly` for the decision graph is returned

Author(s)

Michael Thrun

References

[Wang et al., 2015] Wang, S., Wang, D., Li, C., & Li, Y.: Comment on" Clustering by fast search and find of density peaks", arXiv preprint arXiv:1501.04267, 2015.

[Pedersen et al., 2017] Thomas Lin Pedersen, Sean Hughes and Xiaojie Qiu: densityClust: Clustering by Fast Search and Find of Density Peaks. R package version 0.3. https://CRAN.R-project.org/package=densityClust, 2017.

[Rodriguez/Laio, 2014] Rodriguez, A., & Laio, A.: Clustering by fast search and find of density peaks, Science, Vol. 344(6191), pp. 1492-1496. 2014.

See Also

`ADPclustering`

`densityClust`

Examples

```
data(Hepta)
H=EntropyOfDataField(Hepta$Data,  seq(from=0,to=1.5,by=0.05),PlotIt=FALSE)
Sigmamin=names(H)[which.min(H)]
Dc=3/sqrt(2)*as.numeric(names(H)[which.min(H)])
#look at the plot and estimate rho and delta

DensityPeakClustering(Hepta$Data, Knn = 7,Dc=Dc)
Cls=DensityPeakClustering(Hepta$Data,Dc=Dc,Rho = 0.028,

Delta = 22,Knn = 7,PlotIt = TRUE)$Cls
```

DivisiveAnalysisClustering

Large DivisiveAnalysisClustering Clustering

Description

Divisive Analysis Clustering (diana) of [Rousseeuw/Kaufman, 1990]

Usage

```
DivisiveAnalysisClustering(DataOrDistances, ClusterNo,

PlotIt=FALSE,Standardization=TRUE,Data,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm. if ClusterNo=0, the dendrogram is generated instead of a clustering to estimate the numbers of clusters.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
Standardization	DataOrDistances is standardized before calculating the dissimilarities. Measurements are standardized for each variable (column), by subtracting the variable's mean value and dividing by the variable's mean absolute deviation.If DataOrDistances is already a distance matrix, then this argument will be ignored.
Data	[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial matching does not work.
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Rousseeuw/Kaufman, 1990] Rousseeuw, P. J., & Kaufman, L.: Finding groups in data, Belgium, John Wiley & Sons Inc., ISBN: 0471735787, 1990.

Examples

```
data('Hepta')
out=DivisiveAnalysisClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

print(out$dianaObject)
plot(out$dianaObject)
```

EngyTime *EngyTime of [Baggenstoss, 2002].*

Description

Gaussian mixture. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("EngyTime")
```

Details

Size 4096, Dimensions 2, stored in `EngyTime$Data`

Classes 2, stored in `EngyTime$Cls`

References

[Baggenstoss, 2002] Baggenstoss, P. M.: Statistical modeling using gaussian mixtures and hmms with matlab, Naval Undersea Warfare Center, Newport RI, 2002.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(EngyTime)
str(EngyTime)
```

`EntropyOfDataField` *Entropy Of a Data Field [Wang et al., 2011].*

Description

Calculates the Potential Entropy Of a Data Field for a givven ranges of impact factors sigma

Usage

```
EntropyOfDataField(Data,

sigmarange = c(0.01, 0.1, 0.5, 1, 2, 5, 8, 10, 100)

, PlotIt = TRUE)
```

Arguments

`Data`	[1:n,1:d] data matrix
`sigmarange`	numeric vector [1:s] of relevant sigmas
`PlotIt`	FALSE: disable plot, TRUE: Plot with upper boundary of H after [Wang et al., 2011].

Details

In theory there should be a courve with a clear minimum of Entropy [Wang et al.,2011]. Then the choice for the impact factor sigma is the minimum of the entropy to defined the correct data field. It follows, that the influence radius is 3/sqrt(2)*sigma (3B rule of gaussian distribution) for clustering algorithms like Density Peak clustering [Wang et al.,2011].

Value

1:s named vector of the Entropy of data field. The names are the impact factor sigma

Author(s)

Michael Thrun

References

[Wang et al., 2015] Wang, S., Wang, D., Li, C., & Li, Y.: Comment on" Clustering by fast search and find of density peaks", arXiv preprint arXiv:1501.04267, 2015.

[Wang et al., 2011] Wang, S., Gan, W., Li, D., & Li, D.: Data field for hierarchical clustering, International Journal of Data Warehousing and Mining (IJDWM), Vol. 7(4), pp. 43-63. 2011.

Examples

```
data(Hepta)
H=EntropyOfDataField(Hepta$Data,PlotIt=FALSE)
Sigmamin=names(H)[which.min(H)]
Dc=3/sqrt(2)*as.numeric(names(H)[which.min(H)])
```

`EstimateRadiusByDistance` *Estimate Radius By Distance*

Description

Published in [Thrun et al, 2016] for the case of automatically estimating the radius of the P-matrix. Can also be used to estimate the radius parameter for distance based clustering algorithms.

Usage

```
EstimateRadiusByDistance(DistanceMatrix)
```

Arguments

`DistanceMatrix` [1:n,1:n] Distance Matrix of n cases

Details

For density-based clustering algorithms like `DBscan` it is not always usefull.

Value

Numerical scalar defining the radius

Note

Symmetric matrix is assumed.

Author(s)

Michael Thrun

References

[Thrun et al., 2016] Thrun, M. C., Lerch, F., Loetsch, J., & Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Vol. 24, pp. 7-16, Plzen, http://wscg.zcu.cz/wscg2016/short/A43-full.pdf, 2016.

See Also

`GeneratePmatrix`

Examples

```
data('Hepta')
DistanceMatrix=as.matrix(parallelDist::parallelDist(Hepta$Data))
Radius=EstimateRadiusByDistance(DistanceMatrix)
```

FannyClustering *Fuzzy Analysis Clustering [Rousseeuw/Kaufman, 1990, p. 164-198]*

Description

...

Usage

```
FannyClustering(DataOrDistances,ClusterNo,

PlotIt=FALSE,Standardization=TRUE,Data,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
Standardization	DataOrDistances is standardized before calculating the dissimilarities. Measurements are standardized for each variable (column), by subtracting the variable's mean value and dividing by the variable's mean absolute deviation.If DataOrDistances is already a distance matrix, then this argument will be ignored.
Data	[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial matching does not work.
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

...

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering. . Points which cannot be assigned to a cluster will be reported with 0.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Rousseeuw/Kaufman, 1990] Rousseeuw, P. J., & Kaufman, L.: Finding groups in data, Belgium, John Wiley & Sons Inc., ISBN: 0471735787, 1990.

Examples

```
data('Hepta')
out=FannyClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

GenerateFundamentalClusteringProblem

Generates a Fundamental Clustering Problem based on specific artificial datasets.

Description

Lsun3D and FCPS datasets were introduced in various publications for a specific fixed size. This function generalites them for any sample size.

Usage

```
GenerateFundamentalClusteringProblem(Name,SampleSize,PlotIt=TRUE)
```

Arguments

Name	string, either 'Atom', 'Chainlink, 'EngyTime', 'GolfBall', 'Hepta', 'Lsun3D', 'Target' 'Tetra' 'TwoDiamonds' 'WingNut
SampleSize	Size of Sample higher than 300, preferable above 500
PlotIt	TRUE: Plots the Problem

Details

A detailed description of the datasets can be found in [Thrun, 2018]. Lsun was extended to Lsun3D in [Thrun, 2018]. Sampling works by combining Pareto Density Estimation with rejection sampling.

Value

LIST, with

Name	[1:SampleSize,1:d] data matrix
Cls	[1:SampleSize] numerical vector of classification

Author(s)

Michael Thrun

References

[Thrun, 2018] Thrun, M. C.: Projection Based Clustering through Self-Organization and Swarm Intelligence, Springer, Heidelberg, ISBN: 978-3-658-20539-3, https://doi.org/10.1007/978-3-658-20540-9, 2018.

Examples

```
GenerateFundamentalClusteringProblem("Chainlink",2000,TRUE)
```

GenieClustering *Genie Clustering by Gini Index*

Description

Outlier Resistant Hierarchical Clustering Algorithm of [Gagolewski/Bartoszuk, 2016].

Usage

```
GenieClustering(DataOrDistances, ClusterNo = 0,
DistanceMethod="euclidean", ColorTreshold = 0,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
DistanceMethod	see parDist, for example 'euclidean','mahalanobis','manhatten' (cityblock),'fJaccard','binary', 'canberra', 'maximum'. Any unambiguous substring can be given.
ColorTreshold	draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
...	furter argument to genie like: thresholdGini single numeric value in [0,1], threshold for the Gini index, 1 gives the standard single linkage algorithm

Details

Wrapper for Genie algorithm.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Dendrogram`	Dendrogram of hclust

Author(s)

Michael Thrun

References

[Gagolewski/Bartoszuk, 2016] Gagolewski M., Bartoszuk M., Cena A., Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm, Information Sciences, Vol. 363, pp. 8-23, 2016.

See Also

`HierarchicalClustering`

Examples

```
data('Hepta')
out=GenieClustering(Hepta$Data,ClusterNo=7)
```

GolfBall *GolfBall of [Ultsch, 2005]*

Description

no clusters at all. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("GolfBall")
```

Details

Size 4002, Dimensions 3, stored in `GolfBall$Data`

Classes 1, stored in `GolfBall$Cls`

References

[Ultsch, 2005] Ultsch, A.: Clustering wih SOM: U* C, Proc. Proceedings of the 5th Workshop on Self-Organizing Maps, Vol. 2, pp. 75-82, 2005.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(GolfBall)
str(GolfBall)
```

GraphBasedClustering *MST-kNN clustering algorithm*

Description

Performs the MST-kNN clustering algorithm which generate a clustering solution with automatic k determination using two proximity graphs: Minimal Spanning Tree (MST) and k-Nearest Neighbor (kNN) which are recursively intersected.

Usage

```
GraphBasedClustering(DataOrDistances, method = "euclidean",PlotIt=FALSE, ...)
```

Arguments

DataOrDistances	Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] not symmetric data matrix of n cased and d variable
method	Optional distance method of data, default is euclid, see parDist for details
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Optional, further arguments for mst.knn

Details

Does not work on Hepta with euclidean distances.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

Inostroza-Ponta, M.: An Integrated and Scalable Approach Based on Combinatorial Optimization Techniques for the Analysis of Microarray Data. Ph.D. thesis, School of Electrical Engineering and Computer Science. University of Newcastle, 2008.

See Also

mst.knn

Examples

```
data(Hepta)

GraphBasedClustering(Hepta$Data)
```

HCLclustering *On-line Update (Hard Competitive learning) method*

Description

Hard Competitive learning clustering published by [Ripley, 2007] and implemented by [Dimitriadou, 2002].

Usage

```
HCLclustering(Data, ClusterNo,PlotIt=FALSE,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Dimitriadou, 2002] Dimitriadou, E.: cclust-convex clustering methods and clustering indexes. R package, 2002,

[Ripley, 2007] Ripley, B. D.: Pattern recognition and neural networks, Cambridge university press, ISBN: 0521717701, 2007.

Examples

```
data('Hepta')
out=HCLclustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

Hepta *Hepta of [Ultsch, 2003]*

Description

clearly defined clusters, different variances. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Hepta")
```

Details

Size 212, Dimensions 3, stored in `Hepta$Data`

Classes 7, stored in `Hepta$Cls`

References

[Ultsch, 2003] Ultsch, A.: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, 2003.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Hepta)
str(Hepta)
```

`HierarchicalClusterData` *Hierarchical Clusterering*

Description

Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.Used stats package function 'hclust'.

Usage

```
HierarchicalClusterData(Data,ClusterNo=0,

method="ward.D2",DistanceMethod="euclidean",

ColorTreshold=0,Fast=FALSE,Cls=NULL,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`method`	Methode der Clusterung: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".
`DistanceMethod`	see `parDist`, for example 'euclidean','mahalanobis','manhatten' (cityblock),'fJaccard','binary', 'canberra', 'maximum'. Any unambiguous substring can be given.
`ColorTreshold`	draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
`Fast`	if TRUE and fastcluster installed, then a faster implementation of the methods above can be used
`Cls`	[1:n] classification vector for coloring of dendrogram in plot
`...`	If ClusterNo=0, plot arugments for as.dendrogramm, e.g.leaflab

Details

leaflab : a string specifying how leaves are labeled. The default "perpendicular" write text vertically (by default). "textlike" writes text horizontally (in a rectangle), and "none" suppresses leaf labels s. ?as.dendrogramm

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Dedrogram	Dendrogram of hclust

Author(s)

Michael Thrun

Examples

```
data('Hepta')
out=HierarchicalClusterData(Hepta$Data,ClusterNo=7)
```

HierarchicalClusterDists

HierarchicalClusterDists(pDist) HierarchicalClusterDists(pDist,0,"ward.D",100) Cls=HierarchicalClusterDists(pDist,6,"ward.D") Zeichnet entweder ein Dendrogram oder liefert eine Klassenzuweisung

---

Description

Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.Used stats package function 'hclust'.

Usage

```
HierarchicalClusterDists(pDist,ClusterNo=0,method="ward.D2",

ColorTreshold=0,Fast=FALSE,...)
```

Arguments

pDist	distances as either matrix [1:n,1:n] or dist object
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
method	method of cluster analysis: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".
ColorTreshold	draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
Fast	if TRUE and fastcluster installed, then a faster implementation of the methods above can be used
...	If ClusterNo=0, plot arugments for as.dendrogramm, e.g. leaflab

Details

leaflab : a string specifying how leaves are labeled. The default "perpendicular" write text vertically (by default). "textlike" writes text horizontally (in a rectangle), and "none" suppresses leaf labels s. ?as.dendrogramm

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Dedrogram`	Dendrogram of hclust

Author(s)

Michael Thrun

Examples

```
data('Hepta')
out=HierarchicalClusterData(Hepta$Data,ClusterNo=7)
```

HierarchicalClustering

Hierarchical Clustering

Description

Wrapper various agglomerative hierarchical clustering algorithms.

Usage

```
HierarchicalClustering(DataOrDistances,ClusterNo,method='SingleL',Fast=TRUE,Data,...)
```

Arguments

`DataOrDistances`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`method`	method of cluster analysis: "Ward", "SingleL", "CompleteL", "AverageL" (UPGMA), "WPGMA" (mcquitty), "MedianL" (WPGMC), "CentroidL" (UPGMC), "Minimax", "MinEnergy" or "Gini".
`Fast`	if TRUE and fastcluster installed, then a faster implementation of the methods above can be used except "Minimax", "MinEnergy" or "Gini"

Data	[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial matching does not work.
...	Further arguments passed on to either HierarchicalClusterData, HierarchicalClusterDists, MinimalEnergyClustering or GenieClustering (for "Gini").

Details

Please see HierarchicalClusterData and HierarchicalClusterDists.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Dedrogram	Dendrogram of hclust

Author(s)

Michael Thrun

See Also

HierarchicalClusterData

HierarchicalClusterDists,

MinimalEnergyClustering.

Examples

```
data('Hepta')
out=HierarchicalClustering(Hepta$Data,ClusterNo=7)
```

Hierarchical_DBSCAN *Hierarchical DBSCAN*

Description

Hierarchical DBSCAN clustering [Campello et al., 2015].

Usage

```
Hierarchical_DBSCAN(Data,minPts=4,

PlotTree=FALSE,PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`minPts`	classic smoothing factor in density estimates [Campello et al., 2015, p.9]
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`PlotTree`	default: FALSE, If TRUE plots the dendrogram. If minPts is missing, PlotTree is set to TRUE.
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

"computes the hierarchical cluster tree representing density estimates along with the stability-based flat cluster extraction proposed by Campello et al. (2013). HDBSCAN essentially computes the hierarchy of all DBSCAN* clusterings, and then uses a stability-based extraction method to find optimal cuts in the hierarchy, thus producing a flat solution."[Hahsler et al., 2019]

It is claimed by the inventors that the minPts parameter is noncritical [Campello et al., 2015, p.35]. minPts is reported to be set to 4 on all experiments [Campello et al., 2015, p.35].

Value

List of

`Cls`	[1:n] numerical vector defining the clustering; this classification is the main output of the algorithm. Points which cannot be assigned to a cluster will be reported as members of the noise cluster with 0.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Campello et al., 2015] Campello, R. J., Moulavi, D., Zimek, A., & Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 10(1), pp. 1-51. 2015.

[Hahsler et al., 2019] Hahsler M, Piekenbrock M, Doran D: dbscan: Fast Density-Based Clustering with R. Journal of Statistical Software, 91(1), pp. 1-30. doi: 10.18637/jss.v091.i01, 2019

Examples

```
data('Hepta')

out=Hierarchical_DBSCAN(Hepta$Data,PlotIt=FALSE)
```

`InterClusterDistances` *InterClusterDistances*

Description

Computes inter-cluster distances which are the distance between each cluster and all other clusters

Usage

```
InterClusterDistances(FullDistanceMatrix, Cls,

Names,PlotIt=FALSE)
```

Arguments

`FullDistanceMatrix`	[1:n,1:n] symmetric distance matrix
`Cls`	[1:n] numerical vector of k classes
`Names`	Optional [1:k] character vector naming k classes
`PlotIt`	Optional, Plots if TRUE

Details

Cluster distances are given back as a matrix, one column per cluster and the vector of the full distance matrix without the diagonal elements and the upper half of the symmetric matrix.

Value

matrix [1:m,1:(k+1)] of k clusters, each columns consists of the distances between a cluster and all other clusters, filled up with NaN at the end to be of the same lenght as the complete distance matrix.

Note

PlotIt is under development

Author(s)

Michael Thrun

References

[Thrun, 2018] Thrun, M.C., Projection Based Clustering through Self-Organization and Swarm Intelligence. 2018, Heidelberg: Springer.

See Also

`MDplot`

Examples

```
##ToDo
```

`kmeansClustering` *K-Means Clustering*

Description

Perform k-means clustering on a data matrix. Uses either stats package function 'kmeans' or cclust package implemention.

Usage

```
kmeansClustering(Data, ClusterNo,Centers=NULL,

 method = 'LBG',PlotIt=FALSE, Verbose = FALSE,... )
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`Centers`	default(NULL) a set of initial (distinct) cluster centres.
`method`	Choice of Kmeans algorithm, currently either "Hartigan" or "LBG"
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`Verbose`	print details, if true
`...`	Further arguments like `iter.max`,`nstart`,...

Value

List V of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	List V of SumDistsToCentroids: Vector of within-cluster sum of squares, one component per cluster Centroids: the final cluster centers.

Author(s)

Alfred Ultsch, Michael Thrun

References

Hartigan, J. A. and Wong, M. A.. A K-means clustering algorithm. Applied Statistics 28, 100-108, 1979.

Linde,Y.,Buzo,A.,Gray,R.M., An algorithm for vector quantizer design. IEEE Transactions on Communications, COM-28, 84-95, 1980

Examples

```
data('Hepta')
out=kmeansClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

LargeApplicationClustering

Large Application Clustering

Description

Clustering Large Applications (clara) of [Rousseeuw/Kaufman, 1990]

Usage

```
LargeApplicationClustering(Data, ClusterNo,

PlotIt=FALSE,Standardization=TRUE,Samples=50,Random=TRUE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`Standardization`	`Data` is standardized before calculating the dissimilarities. Measurements are standardized for each variable (column), by subtracting the variable's mean value and dividing by the variable's mean absolute deviation.
`Samples`	integer, say N, the number of samples to be drawn from the dataset. Default value set as recommended by documentation of `clara`
`Random`	logical indicating if R's random number generator should be used instead of the primitive clara()-builtin one.
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

It is recommended to use `set.seed` if clustering output should be always the same instead of setting Random=FALSE in order to use the primitive clara()-builtin random number generator.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Rousseeuw/Kaufman, 1990] Rousseeuw, P. J., & Kaufman, L.: Finding groups in data, Belgium, John Wiley & Sons Inc., ISBN: 0471735787, 1990.

Examples

```
data('Hepta')
out=LargeApplicationClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

Leukemia *Leukemia Distancematrix and classificiation used in [Thrun, 2018]*

Description

Data is anonymized. Original dataset was pubished in [Haferlach et al., 2010]. Original dataset had around 12.000 Dimensions. Detailed descriptions of preprocessed dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Leukemia")
```

Details

554x554 distance matrix. Cls defines the following clusters:

1= APL Outlier

2=APL

3=Healthy

4=AML

5=CLL

6=CLL Outlier

References

[Thrun, 2018] Thrun, M. C.: Projection Based Clustering through Self-Organization and Swarm Intelligence, doctoral dissertation 2017, Springer, Heidelberg, ISBN: 978-3-658-20539-3, https://doi.org/10.1007/978-3-658-20540-9, 2018.

[Haferlach et al., 2010] Haferlach, T., Kohlmann, A., Wieczorek, L., Basso, G., Te Kronnie, G., Bene, M.-C., . . . Mills, K. I.: Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group, Journal of Clinical Oncology, Vol. 28(15), pp. 2529-2537. 2010.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Leukemia)
str(Leukemia)
Cls=Leukemia$Cls
Distance=Leukemia$DistanceMatrix
isSymmetric(Distance)
```

`Lsun` *Lsun from FCPS*

Description

different variances and inter cluster distances

Usage

```
data("Lsun")
```

Details

Size 400, Dimensions 2, stored in `Lsun$Data`

Classes 3, stored in `Lsun$Cls`

References

Ultsch, A.: Clustering with SOM: U*C, In Proc. Workshop on Self-Organizing Maps, Paris, France, (2005) , pp. 75-82

Examples

```
data(Lsun)
str(Lsun)
```

Lsun3D *Lsun3D inspired by FCPS*

Description

clearly defined clusters, different variances. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Lsun3D")
```

Details

Size 404, Dimensions 3

Dataset defined discontinuites, where the clusters have different variances. Three main Clusters, and four Outliers (in Cluster 4). See for a more detailed description in [Thrun, 2018].

References

[Thrun, 2018] Thrun, M. C.: Projection Based Clustering through Self-Organization and Swarm Intelligence, doctoral dissertation 2017, Springer, Heidelberg, ISBN: 978-3-658-20539-3, https://doi.org/10.1007/978-3-658-20540-9, 2018.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Lsun3D)
str(Lsun3D)
Cls=Lsun3D$Cls
Data=Lsun3D$Data
```

MarkovClustering *Markov Clustering*

Description

Graph clustering algorithm introduced by [van Dongen, 2000].

Usage

```
MarkovClustering(Data=NULL,Adjacency=NULL,Radius=TRUE,addLoops =TRUE,PlotIt=FALSE,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. This is used if Adjecency is missing. Then a unit-disk (R-ball) graph is calculated.
Adjacency	Used if Data is missing, matrix [1:n,1:n] defining which points are adjacent to each other by the number 1; not adjacent: 0
Radius	Radius for unit disk graph (r-ball graph) if adjacency matrix is missing. Automatic estimation can be done either with =TRUE [Ultsch, 2005] or FALSE [Thrun et al., 2016]
addLoops	logical; if TRUE, self-loops with weight 1 are added to each vertex of x (see mcl of CRAN package MCL).
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

...

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering. . Points which cannot be assigned to a cluster will be reported with 0.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[van Dongen, 2000] van Dongen, S.M. Graph Clustering by Flow Simulation. Ph.D. thesis, Universtiy of Utrecht. Utrecht University Repository: http://dspace.library.uu.nl/handle/1874/848, 2000

[Thrun et al., 2016] Thrun, M. C., Lerch, F., Loetsch, J., & Ultsch, A. : Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Vol. 24, Plzen, 2016.

[Ultsch, 2005] Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery, In Baier, D. & Werrnecke, K. D. (Eds.), Innovations in classification, data science, and information systems, (Vol. 27, pp. 91-100), Berlin, Germany, Springer, 2005.

Examples

```
data('Hepta')
out=MarkovClustering(Data=Hepta$Data,PlotIt=FALSE)
```

MinimalEnergyClustering

Minimal Energy Clustering

Description

Hierchical Clustering using the minimal energy approach of [Szekely/Rizzo, 2005].

Usage

```
MinimalEnergyClustering(DataOrDistances, ClusterNo = 0,
DistanceMethod="euclidean", ColorTreshold = 0,Data,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
DistanceMethod	see parDist, for example 'euclidean','mahalanobis','manhatten' (cityblock),'fJaccard','binary', 'canberra', 'maximum'. Any unambiguous substring can be given.
ColorTreshold	draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
Data	[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial matching does not work.
...	If ClusterNo=0, plot arugments for as.dendrogramm, e.g. leaflab

Details

leaflab : a string specifying how leaves are labeled. The default "perpendicular" write text vertically (by default). "textlike" writes text horizontally (in a rectangle), and "none" suppresses leaf labels s. ?as.dendrogramm

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Dendrogram`	Dendrogram of hclust

Author(s)

Michael Thrun

References

[Szekely/Rizzo, 2005] Szekely, G. J. and Rizzo, M. L.: Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method, Journal of Classification, 22(2) 151-183.http://dx.doi.org/10.1007/s00357-005-0012-9, 2005.

See Also

`HierarchicalClustering`

Examples

```
data('Hepta')
out=MinimalEnergyClustering(Hepta$Data,ClusterNo=7)
```

MinimaxLinkageClustering

Minimax Linkage Hierarchical Clustering

Description

inimax linkage hierarchical clustering. Every cluster has an associated prototype element that represents that cluster [Bien/Tibshirani, 2011].

Usage

```
MinimaxLinkageClustering(DataOrDistances, ClusterNo = 0,
DistanceMethod="euclidean", ColorTreshold = 0,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
DistanceMethod	see parDist, for example 'euclidean','mahalanobis','manhatten' (cityblock),'fJaccard','binary', 'canberra', 'maximum'. Any unambiguous substring can be given.
ColorTreshold	draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
...	If ClusterNo=0, plot arugments for as.dendrogramm, e.g. leaflab

Details

Wrapper for Minimax Linkage Hierarchical Clustering algorithm.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Dendrogram	Dendrogram of hclust, if ClusterNo=0
Object	object of protoclust if ClusterNo!=0

Author(s)

Michael Thrun

References

[Bien/Tibshirani, 2011] Bien, J., and Tibshirani, R.: Hierarchical Clustering with Prototypes via Minimax Linkage, The Journal of the American Statistical Association, Vol. 106(495), pp. 1075-1084, 2011.

See Also

HierarchicalClustering

Examples

```
data('Hepta')
out=MinimaxLinkageClustering(Hepta$Data,ClusterNo=7)
```

MinSpanTree *Zeichnet einen 2 dimensionalen minimal spanning Tree*

Description

Clustering with Minimum Spanning Tree

Usage

```
MinSpanTree(DataOrDistance, isDistance)
```

Arguments

DataOrDistance	Der Datensatz oder die Distanzmatrix
isDistance	Setze gleich True falls die Eingabe Distanzen sind

Value

TR an object of class spantree which is a list with two vectors,

ModelBasedClustering *Model Based Clustering*

Description

Calls Model based clustering of [Fraley/Raftery, 2006] which models a Mixture Of Gaussians (MoG).

Usage

```
ModelBasedClustering(Data,ClusterNo=2,PlotIt=FALSE,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

see [Thrun, 2017, p. 23] or [Fraley/Raftery, 2002] and [Fraley/Raftery, 2006].

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Note

MoGclustering used in [Thrun, 2017] was renamed to `ModelBasedClustering` in this package.

Author(s)

Michael Thrun

References

[Thrun, 2017] Thrun, M. C.:A System for Projection Based Clustering through Self-Organization and Swarm Intelligence, (Doctoral dissertation), Philipps-Universitaet Marburg, Marburg, 2017.

[Fraley/Raftery, 2002] Fraley, C., and Raftery, A. E.: Model-based clustering, discriminant analysis, and density estimation, Journal of the American Statistical Association, Vol. 97(458), pp. 611-631. 2002.

[Fraley/Raftery, 2006] Fraley, C., and Raftery, A. E.MCLUST version 3: an R package for normal mixture modeling and model-based clustering,DTIC Document, 2006.

See Also

`MoGclustering`

Examples

```
data('Hepta')
out=ModelBasedClustering(Hepta$Data,PlotIt=FALSE)
```

`MoGclustering` *MoGclustering*

Description

call MixtureOfGaussians (MoG) clustering based on Expectation Maximization (EM)

Usage

```
MoGclustering(Data,ClusterNo=2,method="EM",PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`method`	Initialization by either "EM" oder "kmeans"
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

...

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Note

MoG used in [Thrun, 2017] was renamed to `ModelBasedClustering` in this package.

Author(s)

Michael Thrun

References

[Thrun, 2017] Thrun, M. C.:A System for Projection Based Clustering through Self-Organization and Swarm Intelligence, (Doctoral dissertation), Philipps-Universitaet Marburg, Marburg, 2017.

See Also

`ModelBasedClustering`

Examples

```
data('Hepta')
out=MoGclustering(Hepta$Data,PlotIt=FALSE)
```

`NeuralGasClustering` *Neural gas algorithm for clustering*

Description

Neural gas clustering published by [Martinetz et al., 1993]] and implemented by [Bodenhofer et al., 2011].

Usage

```
NeuralGasClustering(Data, ClusterNo,PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Dimitriadou, 2002] Dimitriadou, E.: cclust-convex clustering methods and clustering indexes. R package, 2002,

[Martinetz et al., 1993] Martinetz, T. M., Berkovich, S. G., & Schulten, K. J.: 'Neural-gas' network for vector quantization and its application to time-series prediction, IEEE Transactions on Neural Networks, Vol. 4(4), pp. 558-569. 1993.

Examples

```
data('Hepta')
out=NeuralGasClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

OPTICSclustering *OPTICS Clustering*

Description

OPTICS (Ordering points to identify the clustering structure) clustering algorithm [Ankerst et al.,1999].

Usage

```
OPTICSclustering(Data, MaxRadius,RadiusThreshold, minPts = 5, PlotIt=FALSE,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
MaxRadius	upper limit neighborhood in the R-ball graph/unit disk graph), size of the epsilon neighborhood (eps) [Ester et al., 1996, p. 227]. If missing, automatic estimation is done using insights of [Ultsch, 2005].
RadiusThreshold	Threshold to identify clusters (RadiusThreshold <= MaxRadius), if not given `0.9*MaxRadius` is set.
minPts	number of minimum points in the eps region (for core points). In principle minimum number of points in the unit disk, if the unit disk is within the cluster (core) [Ester et al., 1996, p. 228]. Default is 2.5 percent of points.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

...

Value

List of

Cls	[1:n] numerical vector defining the clustering; this classification is the main output of the algorithm. Points which cannot be assigned to a cluster will be reported as members of the noise cluster with 0.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Ankerst et al.,1999] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Joerg Sander: OPTICS: Ordering Points To Identify the Clustering Structure, ACM SIGMOD international conference on Management of data, ACM Press, pp. 49-60, 1999.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, Proc. Kdd, Vol. 96, pp. 226-231, 1996.

[Ultsch, 2005] Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery, In Baier, D. & Werrnecke, K. D. (Eds.), Innovations in classification, data science, and information systems, (Vol. 27, pp. 91-100), Berlin, Germany, Springer, 2005.

See Also

optics

Examples

```
data('Hepta')
out=OPTICSclustering(Hepta$Data,PlotIt = FALSE)
```

PAMclustering *Partitioning Around Medoids (PAM)*

Description

Partitioning (clustering) of the data into k clusters around medoids, a more robust version of k-means [Rousseeuw/Kaufman, 1990, p. 164-198] .

Usage

```
PAMclustering(DataOrDistances,ClusterNo,

PlotIt=FALSE,Standardization=TRUE,Data,...)
```

Arguments

DataOrDistances	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls

`Standardization`
: `DataOrDistances` is standardized before calculating the dissimilarities. Measurements are standardized for each variable (column), by subtracting the variable's mean value and dividing by the variable's mean absolute deviation.If `DataOrDistances` is already a distance matrix, then this argument will be ignored.

`Data`
: [1:n,1:d] data matrix in the case that `DataOrDistances` is missing and partial matching does not work.

`...`
: Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

[Rousseeuw/Kaufman, 1990, chapter 2] or [Reynolds et al., 1992].

Value

List of

`Cls`
: [1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.

`Object`
: Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Rousseeuw/Kaufman, 1990] Rousseeuw, P. J., & Kaufman, L.: Finding groups in data, Belgium, John Wiley & Sons Inc., ISBN: 0471735787, 1990.

[Reynolds et al., 1992] Reynolds, A., Richards, G.,de la Iglesia, B. and Rayward-Smith, V.: Clustering rules: A comparison of partitioning and hierarchical clustering algorithms, Journal of Mathematical Modelling and Algorithms 5, 475-504, DOI:10.1007/s10852-005-9022-1, 1992.

Examples

```
data('Hepta')
out=PAMclustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

pdfClustering *Probability Density Distribution Clustering*

Description

Clustering via nonparametric density estimation

Usage

```
pdfClustering(Data, PlotIt = FALSE, ...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

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).

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

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.

Examples

```
data('Hepta')
out=pdfClustering(Hepta$Data,PlotIt=FALSE)
```

`QTclustering` *Stochastic QT Clustering*

Description

stochastic quality clustering of [Heyer et al., 1999] with an improved implementation by [Scharl/Leisch, 2006].

Usage

```
QTclustering(Data,Radius,PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`Radius`	Maximum radius of clusters. Automatic estimation can be done with [Thrun et al., 2016] if not otherwise set.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering. Points which cannot be assigned to a cluster will be reported with 0.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Heyer et al., 1999] Heyer, L. J., Kruglyak, S., & Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes, Genome research, Vol. 9(11), pp. 1106-1115. 1999.

[Scharl/Leisch, 2006] Scharl, T., & Leisch, F.: The stochastic QT-clust algorithm: evaluation of stability and variance on time-course microarray data, in Rizzi , A. & Vichi, M. (eds.), Proc. Proceedings in Computational Statistics (Compstat), pp. 1015-1022, Physica Verlag, Heidelberg, Germany, 2006.

[Thrun et al., 2016] Thrun, M. C., Lerch, F., Loetsch, J., & Ultsch, A. : Visualization and 3D Printing of Multivariate Data of Biomarkers, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Vol. 24, Plzen, 2016.

[Ultsch, 2005] Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery, In Baier, D. & Werrnecke, K. D. (Eds.), Innovations in classification, data science, and information systems, (Vol. 27, pp. 91-100), Berlin, Germany, Springer, 2005.

Examples

```
data('Hepta')
out=QTclustering(Hepta$Data,PlotIt=FALSE)
```

`RobustTrimmedClustering`

Robust Trimmed Clustering Clustering

Description

Robust Trimmed Clustering Clustering of Garcia-Escudero (2008)

Usage

```
RobustTrimmedClustering(Data, ClusterNo,

PlotIt=FALSE,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the algorithm.
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`...`	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

Currently the easies dataset does not work with default parametrization, please see tclust for parameter description.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

Garcia-Escudero, L. A., Gordaliza, A., Matran, C., & Mayo-Iscar, A. (2008). A general trimming approach to robust cluster analysis. The Annals of Statistics, 36(3), 1324-1345.

Fritz, H., Garcia-Escudero, L. A., & Mayo-Iscar, A. (2012). tclust: An R package for a trimming approach to cluster analysis. Journal of Statistical Software, 47(12), 1-26.

Examples

```
data('Hepta')
out=RobustTrimmedClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE,restr.fact=1e3)
```

SharedNearestNeighborClustering

SNN clustering

Description

Shared Nearest Neighbor Clustering of [Ertoz et al., 2003].

Usage

```
SharedNearestNeighborClustering(Data,Knn=7,

Radius,minPts,PlotIt=FALSE,

UpperLimitRadius,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
Knn	number of neighbors to consider to calculate the shared nearest neighbors.
Radius	eps [Ester et al., 1996, p. 227] neighborhood in the R-ball graph/unit disk graph), size of the epsilon neighborhood. If missing, automatic estimation is done using insights of [Ultsch, 2005].
minPts	number of minimum points in the eps region (for core points). In principle minimum number of points in the unit disk, if the unit disk is within the cluster (core) [Ester et al., 1996, p. 228]. Default is 2.5 percent of points.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
UpperLimitRadius	Limit for radius search, experimental
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

Details

..

Value

List of

Cls	[1:n] numerical vector defining the clustering; this classification is the main output of the algorithm. Points which cannot be assigned to a cluster will be reported as members of the noise cluster with 0.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Ertoz et al., 2003] Levent Ertoz, Michael Steinbach, Vipin Kumar: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data, SIAM International Conference on Data Mining, 47-59, 2003.

See Also

sNNclust

Examples

```
data('Hepta')
out=SharedNearestNeighborClustering(
Hepta$Data,PlotIt = FALSE)
```

`SOMclustering` *self-organizing maps based clustering implemented by [Wherens, Buydens, 2017].*

Description

Either the variant k-batch or k-online is possible in which every unit can be seen approximatly as an cluster.

Usage

```
SOMclustering(Data,LC=c(1,2),ClusterNo=NULL,

Mode="online",PlotIt=FALSE,rlen=100,alpha = c(0.05, 0.01),...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`LC`	Lines and Columns of a very small SOM, usually every unit is a cluster, will bhe ignored if ClusterNo is not NULL.
`ClusterNo`	Optional, A number k which defines k different Clusters to be build by the algorithm. LC will then be set accordingly.
`Mode`	either "batch" or "online"
`PlotIt`	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`
`rlen`	Please see `supersom`
`alpha`	Please see `supersom`
`...`	Further arguments to be set for the clustering algorithm in `somgrid`, if not set, default arguments are used.

Details

This clustering algorithm is based on very small maps and, hence, not emergent (c.f. [Thrun, 2018, p.37]). A 3x3 map means 9 units leading to 9 clusters.

Batch is a deterministic clustering approach whereas online ist stochastic clustering approach and research indicates that online should be prefereeed (c.f. [Thrun, 2018, p.37]).

Value

List of

`Cls`	[1:n] numerical vector defining the classification as the main output of the clustering algorithm
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[Wherens, Buydens, 2017] R. Wehrens and L.M.C. Buydens, J. Stat. Softw. 21 (5), 2007; R. Wehrens and J. Kruisselbrink, submitted, 2017.

[Thrun, 2018] Thrun, M.C., Projection Based Clustering through Self-Organization and Swarm Intelligence. 2018, Heidelberg: Springer.

Examples

```
data('Hepta')
out=SOMclustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

SpectralClustering *Spectral Clustering*

Description

Clusters the Data into "ClusterNo" different clusters using the Spectal Clustering Method

Usage

```
SpectralClustering(Data, ClusterNo,PlotIt=FALSE,...)
```

Arguments

Data	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
ClusterNo	A number k which defines k different Clusters to be build by the algorithm.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
...	Further arguments to be set for the clustering algorithm, if not set, default arguments are used. e.g.: kernel : Kernelmethod, possible options: rbfdot Radial Basis kernel function "Gaussian" polydot Polynomial kernel function vanilladot Linear kernel function tanhdot Hyperbolic tangent kernel function laplacedot Laplacian kernel function besseldot Bessel kernel function anovadot ANOVA RBF kernel function splinedot Spline kernel stringdot String kernel kpar : Kernelparameter: a character string or the list of hyper-parameters (kernel parameters). The default character string "automatic" uses a heuristic to determine a suitable value for the width parameter of the RBF kernel. "local" (local scaling) uses a more advanced heuristic and sets a width parameter for every point in the data set. A list can also be used containing the parameters to be used with the kernel function.

Value

List of

Cls	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
Object	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

Examples

```
data('Hepta')
out=SpectralClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

Spectrum *Fast Adaptive Spectral Clustering [John et al, 2020]*

Description

Spectrum is a self-tuning spectral clustering method for single or multi-view data. In this wrapper restricted to the standard used in other clustering algorithms.

Usage

```
Spectrum(Data, Method = 2, ClusterNo = NULL,

PlotIt = FALSE, Silent = TRUE,PlotResults = FALSE, ...)
```

Arguments

Data	1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
Method	Method=1: default eigengap method (Gaussian clusters) Method=2: multimodality gap method (Gaussian/ non-Gaussian clusters) Method=3: Allows to setClusterNo
ClusterNo	Optional, A number k which defines k different Clusters to be build by the algorithm. For default ClusterNo=NULL please see details.
PlotIt	default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in Cls
Silent	silent progress of algorithm=TRUE
PlotResults	plots result of spectrum with plot function
...	method: Numerical value: 1 = default eigengap method (Gaussian clusters), 2 = multimodality gap method (Gaussian/ non-Gaussian clusters), 3 = no automatic method (see fixk param) other paras defined in Spectrum packages

Details

Spectrum is a Partitioning algorithm and either uses the eigengap or multimodality gap heuristics to determine the number of clusters, please see Spectrum package dor details

Value

List of

`Cls`	[1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.
`Object`	Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References

[John et al, 2020] John, C. R., Watson, D., Barnes, M. R., Pitzalis, C., & Lewis, M. J.: Spectrum: Fast density-aware spectral clustering for single and multi-omic data. Bioinformatics, Vol. 36(4), pp. 1159-1166, 2020.

See Also

`Spectrum`

Examples

```
data('Hepta')
out=Spectrum(Hepta$Data,PlotIt=FALSE)

out=Spectrum(Hepta$Data,PlotIt=TRUE)
```

`StatPDEdensity` *Pareto Density Estimation*

Description

Density Estimation for ggplot with a clear model behind it.

Format

The format is: Classes 'StatPDEdensity', 'Stat', 'ggproto' <ggproto object: Class StatPDEdensity, Stat> aesthetics: function compute_group: function compute_layer: function compute_panel: function default_aes: uneval extra_params: na.rm finish_layer: function non_missing_aes: parameters: function required_aes: x y retransform: TRUE setup_data: function setup_params: function super: <ggproto object: Class Stat>

Details

PDE was published in [Ultsch, 2005], short explanation in [Thrun, Ultsch 2018] and the PDE optimized violin plot was published in [Thrun et al., 2018].

References

[Ultsch,2005] Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery, in Baier, D.; Werrnecke, K. D., (Eds), Innovations in classification, data science, and information systems, Proc Gfkl 2003, pp 91-100, Springer, Berlin, 2005.

[Thrun, Ultsch 2018] Thrun, M. C., & Ultsch, A. : Effects of the payout system of income taxes to municipalities in Germany, in Papiez, M. & Smiech,, S. (eds.), Proc. 12th Professor Aleksander Zelias International Conference on Modelling and Forecasting of Socio-Economic Phenomena, pp. 533-542, Cracow: Foundation of the Cracow University of Economics, Cracow, Poland, 2018.

[Thrun et al, 2018] Thrun, M. C., Pape, F., & Ultsch, A. : Benchmarking Cluster Analysis Methods using PDE-Optimized Violin Plots, Proc. European Conference on Data Analysis (ECDA), accepted, Paderborn, Germany, 2018.

`SubspaceClustering` *Algorithms for Subspace clustering*

Description

Subspace clustering is a technique which finds clusters within different subspaces (a selection of one or more dimensions)

Usage

```
SubspaceClustering(Data,ClusterNo,DimSubspace,

method='Orclus',PlotIt=FALSE,OrclusInitialClustersNo=ClusterNo+2,...)
```

Arguments

`Data`	[1:n,1:d] matrix of dataset to be clustered. It consists of n cases or d-dimensional data points. Every case has d attributes, variables or features.
`ClusterNo`	A number k which defines k different Clusters to be build by the proclus or orclust algorithm.
`DimSubspace`	numerical number defining the dimensionality in which clusters should be search in in the orclust algorithm, for proclus it is an optional parameter
`method`	'Orclus', Subspace Clustering Based on Arbitrarily Oriented Projected Cluster Generation [Aggarwal and Yu, 2000] 'ProClus' ProClus Algorithm for Projected Clustering [Aggarwal/Wolf, 1999] 'Clique' ProClus Algorithm for Projected Clustering [Agrawal/Gehrke et al., 1999] 'SubClu' ProClus Algorithm for Projected Clustering [Kailing et al.,2004]

`PlotIt` default: FALSE, If TRUE plots the first three dimensions of the dataset with colored three-dimensional data points defined by the clustering stored in `Cls`

`OrclusInitialClustersNo`
Only for Orclus algorithm: Initial number of clusters (that are computed in the entire data space). Must be greater than k. The number of clusters is iteratively decreased by factor a until the final number of k clusters is reached.

`...` Further arguments to be set for the clustering algorithm, if not set, default arguments are used.

For Subclue: "epsilon" and "minSupport", see `DBscan`

For Clique: "xi" (number of intervalls for each dimension) and "tau" (Density Threshold), see `DBscan`

Details

"The underlying assumption is that we can find valid clusters which are defined by only a subset of dimensions (it is not needed to have the agreement of all N features).The resulting clusters may be overlapping both in the space of features and observations" [Source: URL].

Value

List of

`Cls` [1:n] numerical vector with n numbers defining the classification as the main output of the clustering algorithm. It has k unique numbers representing the arbitrary labels of the clustering.

`Object` Object defined by clustering algorithm as the other output of this algorithm

Note

JAVA_HOME has to be set for rJava to the ProClus algorithm (in windows set PATH env. variable to .../bin path of Java. The architecture of R and Java have to match. Java automatically downloads the Java version of the browser which may not be installed in the architecture in R. In such a case choose a Java version manually.

Author(s)

Michael Thrun

References

[Aggarwal/Wolf et al., 1999] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S.: Fast algorithms for projected clustering, Proc. ACM SIGMoD Record, Vol. 28, pp. 61-72, ACM, 1999.

[Aggarwal/Yu, 2000] Aggarwal, C. C., & Yu, P. S.: Finding generalized projected clusters in high dimensional spaces, (Vol. 29), ACM, ISBN: 1581132174, 2000.

[Aggarwal/Gehrke et al., 1999]: Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, In Proc. ACM SIGMOD, 1999.

[Kailing et al.,2004] Kailing, Karin, Hans-Peter Kriegel, and Peer Kroeger: Density-connected subspace clustering for high-dimensional data, Proceedings of the 2004 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2004

Further "advertising" can be found in

https://towardsdatascience.com/subspace-clustering-7b884e8fff73

Examples

```
data('Hepta')
out=SubspaceClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```

Target *Target of [Ultsch, 2005].*

Description

Detailed description of dataset and its clustering challenge of outliers is provided in [Thrun/Ultsch, 2020]

Usage

```
data("Target")
```

Details

Size 770, Dimensions 2, stored in `Target$Data`

Classes 6, stored in `Target$Cls`

References

[Ultsch, 2005] Ultsch, A.: U* C: Self-organized Clustering with Emergent Feature Maps, Proc. Lernen, Wissensentdeckung und Adaptivitaet (LWA/FGML), pp. 240-244, Saarbruecken, Germany, 2005.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Target)
str(Target)
```

Tetra *Tetra of [Ultsch, 1993]*

Description

almost touching clusters. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("Tetra")
```

Details

Size 400, Dimensions 3, stored in `Tetra$Data`

Classes 4, stored in `Tetra$Cls`

References

[Ultsch, 1993] Ultsch, A.: Self-organizing neural networks for visualisation and classification, Information and classification, (pp. 307-313), Springer, 1993.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(Tetra)
str(Tetra)
```

TwoDiamonds *TwoDiamonds of [Ultsch, 2003a, 2003b]*

Description

cluster border defined by density. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("TwoDiamonds")
```

Details

Size 800, Dimensions 2, stored in `TwoDiamonds$Data`

Classes 2, stored in `TwoDiamonds$Cls`

References

[Ultsch, 2003a] Ultsch, A.Optimal density estimation in data containing clusters of unknown structure, technical report, Vol. 34,University of Marburg, Department of Mathematics and Computer Science, 2003.

[Ultsch, 2003b] Ultsch, A.: U*-matrix: a tool to visualize clusters in high dimensional data, Fachbereich Mathematik und Informatik, 2003.

[Thrun/Ultsch, 2020] Thrun, M. C., Ultsch, A.: Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems, Data in Brief, 2020.

Examples

```
data(TwoDiamonds)
str(TwoDiamonds)
```

`WingNut` *WingNut of [Ultsch, 2005]*

Description

density vs. distance. Detailed description of dataset and its clustering challenge is provided in [Thrun/Ultsch, 2020].

Usage

```
data("WingNut")
```

Details

Size 1016, Dimensions 2, stored in `WingNut$Data`

Classes 2, stored in `WingNut$Cls`

References

[Ultsch, 2005] Ultsch, A.: Clustering wih SOM: U* C, Proc. Proceedings of the 5th Workshop on Self-Organizing Maps, Vol. 2, pp. 75-82, 2005.

Examples

```
data(WingNut)
str(WingNut)
```

Index