Package ‘FCPS’

May 20, 2022

Type Package
Title Fundamental Clustering Problems Suite
Version 1.3.1
Date 2022-05-20
Maintainer Michael Thrun <m.thrun@gmx.net>

Description Over sixty clustering algorithms are provided in this package with consistent input and output, which enables the user to try out algorithms swiftly. Additionally, 26 statistical approaches for the estimation of the number of clusters as well as the mirrored density plot (MD-plot) of clusterability are implemented. The packages is published in Thrun, M.C., Stier Q.: ”Fundamental Clustering Algorithms Suite” (2021), SoftwareX, <DOI:10.1016/j.softx.2020.100642>. Moreover, the fundamental clustering problems suite (FCPS) offers a variety of clustering challenges any algorithm should handle when facing real world data, see Thrun, M.C., Ultsch A.: ”Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems” (2020), Data in Brief, <DOI:10.1016/j.dib.2020.105501>.

Imports mclust, ggplot2, DataVisualizations
Suggests kernlab, cclust, dbscan, kohonen, MCL, ADPclust, cluster, DatabionicSwarm, orclus, subspace, flexclust, ABCAnalysis, apcluster, pracma,EMCluster, pdfCluster, parallelDist, plotly, ProjectionBasedClustering, GeneralizedUmatrix, mstknclust, densityClust, parallel, energy, R.utils, tclust, Spectrum, genie, protocolust, fastcluster, clusterability, signal, reshape2, PPCI, clustrd, smacof, rgl,prclust, dendextend, moments,prabclus, VarSelLCM, sparcl, mixtools, HDclassif, clustvarsel, yardstick, knitr, rmarkdown, igraph, leiden, clusterSim, NetworkToolbox

Depends R (>= 3.5.0)
License GPL-3
LazyData TRUE
LazyLoad yes

URL https://www.deepbionics.org/

BugReports https://github.com/Mthrun/FCPS/issues
Encoding  UTF-8
VignetteBuilder  knitr
SystemRequirements  Pandoc (>= 1.12.3)
NeedsCompilation  no
Author  Michael Thrun [aut, cre, cph] (<https://orcid.org/0000-0001-9542-5543>),
        Peter Nahrgang [ctr, ctb],
        Felix Pape [ctr, ctb],
        Vasyl Pihur [ctb],
        Guy Brock [ctb],
        Susmita Datta [ctb],
        Somnath Datta [ctb],
        Luis Winckelmann [com],
        Alfred Ultsch [dtc, ctb],
        Quirin Stier [ctb, rev]
Repository  CRAN
Date/Publication  2022-05-20 16:40:05 UTC

R topics documented:

FCPS-package .......................................................... 4
ADPclustering .......................................................... 5
AgglomerativeNestingClustering ..................................... 6
APClustering ............................................................ 8
Atom ........................................................................ 10
AutomaticProjectionBasedClustering .................................. 10
Chainlink .................................................................... 13
ClusterabilityMDplot ..................................................... 14
ClusterApply ............................................................... 16
ClusterChallenge ......................................................... 18
ClusterCount ............................................................... 19
ClusterCreateClassification ........................................... 20
ClusterDaviesBouldinIndex ........................................... 21
ClusterDendrogram ...................................................... 22
ClusterDistances ......................................................... 24
ClusterDunnIndex ......................................................... 25
ClusterEqualWeighting ................................................ 26
ClusteringAccuracy ...................................................... 27
ClusterInterDistances ................................................... 29
ClusterMCC ................................................................ 30
ClusterNoEstimation ..................................................... 31
ClusterNormalize ........................................................ 34
ClusterPlotMDS ........................................................... 35
ClusterRedefine ........................................................... 37
ClusterRename ............................................................. 38
ClusterRenameDescendingSize ....................................... 39
ClusterShannonInfo ....................................................... 40
ClusterUpsamplingMinority .................................................. 41
DBSCAN ................................................................. 43
DBSclusteringAndVisualization ........................................... 45
DensityPeakClustering ...................................................... 48
DivisiveAnalysisClustering ............................................... 50
EngyTime ................................................................. 51
EntropyOfDataField ......................................................... 52
EstimateRadiusByDistance ............................................... 53
FannyClustering ............................................................. 54
GenieClustering ............................................................. 55
GolfBall ..................................................................... 56
HCLclustering ............................................................... 57
HDDClustering .............................................................. 58
Hepta ....................................................................... 59
HierarchicalClusterData .................................................... 60
HierarchicalClusterDists .................................................... 61
HierarchicalClustering ...................................................... 63
HierarchicalDBSCAN ....................................................... 64
kmeansClustering .......................................................... 66
kmeansDist ................................................................. 68
LargeApplicationClustering ................................................. 69
Leukemia ................................................................. 70
Lsun3D ................................................................. 71
MarkovClustering ........................................................... 72
MinimalEnergyClustering ................................................... 74
MinimaxLinkageClustering ............................................... 75
ModelBasedClustering ....................................................... 76
ModelBasedVarSelClustering .............................................. 77
MoGclustering ............................................................... 79
MSTclustering ............................................................... 81
NetworkClustering .......................................................... 82
NeuralGasClustering ......................................................... 83
OPTICScustering .......................................................... 84
PAMclustering ............................................................. 86
pdfClustering .............................................................. 87
PenalizedRegressionBasedClustering .................................... 88
ProjectionPursuitClustering ................................................. 90
QTclustering .............................................................. 91
RobustTrimmedClustering .................................................. 93
SharedNearestNeighborClustering ....................................... 94
SOMclustering ............................................................. 96
SOTAclustering ............................................................ 97
SparseClustering ........................................................... 98
SpectralClustering ........................................................ 100
Spectrum ................................................................. 101
StatPDEdensity ............................................................. 103
SubspaceClustering ......................................................... 103
TandemClustering ........................................................ 105
Description

Over sixty clustering algorithms are provided in this package with consistent input and output, which enables the user to try out algorithms swiftly. Additionally, 26 statistical approaches for the estimation of the number of clusters as well as the mirrored density plot (MD-plot) of clusterability are implemented. The package is published in Thrun, M.C., Stier Q.: "Fundamental Clustering Algorithms Suite" (2021), SoftwareX. <DOI:10.1016/j.softx.2020.100642>. Moreover, the fundamental clustering problems suite (FCPS) offers a variety of clustering challenges any algorithm should handle when facing real world data, see Thrun, M.C., Ultsch A.: "Clustering Benchmark Datasets Exploiting the Fundamental Clustering Problems" (2020), Data in Brief, <DOI:10.1016/j.dib.2020.105501>.

The package consists of many algorithms and fundamental datasets for clustering published in [Thrun/Stier, 2021]. Originally, the 'Fundamental Clustering Problems Suite’ (FCPS) offered a variety of clustering problems any algorithm shall be able to handle when facing real world data. Nine of the here presented artificial datasets were priorly named FCPS with a fixed sample size in Ultsch, A.: "Clustering with SOM: U*C", In Workshop on Self-Organizing Maps, 2005. FCPS often served in the paper as an elementary benchmark for clustering algorithms. The FCPS package extends datasets, enables variable sample sizes for these datasets, and provides a standardized and easy access to many clustering algorithms.

https://www.deepbionics.org/

Details

FCPS datasets consists of data sets with known a priori classification to be reproduced by the algorithms. 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]). The original sample size defined in the respective first publication mentioning the data was used in [Thrun/Ultsch, 2020], but using the R function "ClusterChallenge" (...) any sample size can be drawn for all artificial data sets. [Thrun/Ultsch, 2020]

Index: This package was not yet installed at build time.
ADPclustering

Author(s)

NA

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

References


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 parameters 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 of 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

Details

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

- **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**


**See Also**

- `DensityPeakClustering`
- `adpclust`

**Examples**

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

**AgglomerativeNestingClustering**

**AGNES clustering**

**Description**

Agglomerative hierarchical clustering (AGNES) of [Rousseeuw/Kaufman, 1990, pp. 199-252]

**Usage**

```r
AgglomerativeNestingClustering(DataOrDistances, ClusterNo,
PlotIt = FALSE, Standardization = TRUE, ...)
```
AgglomerativeNestingClustering

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 built by the algorithm. If ClusterNo=0, the dendrogram is generated instead of a clustering to estimate the numbers of clusters.

PlotIt Default: FALSE if codeClusterNo!=0, If TRUE or codeClusterNo=0 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.

... 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.

Dendrogram Dendrogram of hierarchical clustering algorithm

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

Author(s)

Michael Thrun

References


See Also

agnes
Examples

```r
data('Hepta')
CA=AgglomerativeNestingClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
## Not run:
ClusterDendrogram(CA$Dendrogram,7,main='AGNES clustering')

print(CA$Object)
plot(CA$Object)
## End(Not run)
```

---

**APclustering**

*Affinity Propagation Clustering*

---

**Description**

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

**Usage**

```r
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 latter 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.
**APclustering**

**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**

Distance matrix D is converted to similarity matrix S with S = -(D^2).

If data matrix is used, then euclidean similarities are calculated by `similarities` and a specified 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**


Further details in [http://www.bioinf.jku.at/software/apcluster/](http://www.bioinf.jku.at/software/apcluster/)

**See Also**

apcluster

**Examples**

```r
data('Hepta')
res=APclustering(Hepta$Data, PlotIt = FALSE)
```
**AutomaticProjectionBasedClustering**

**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**

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

**Details**

- Size 800, Dimensions 3, stored in `Atom$Data`
- Classes 2, stored in `Atom$Cls`

**References**


**Examples**

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

---

**AutomaticProjectionBasedClustering**

*Automatic Projection-Based Clustering*

**Description**

Projection-based clustering `AutomaticProjectionBasedClustering` projects the data (nonlinear) into two dimensions and tries only to preserve relevant neighborhoods prior to clustering. The cluster analysis itself includes the high-dimensional distances in the clustering process. Performs non-interactive projection-based clustering based on non-linear projection methods [Thrun/Ultsch, 2017], [Thrun/Ultsch, 2020a].
**Usage**

AutomaticProjectionBasedClustering(DataOrDistances, ClusterNo, Type="NerV", StructureType = TRUE, PlotIt=FALSE, PlotTree=FALSE, PlotMap=FALSE,...)

**Arguments**

- **DataOrDistances**
  Either nonsymmetric \([1:n,1:d]\) numerical matrix of a dataset to be clustered. It consists of \(n\) cases of \(d\)-dimensional data points. Every case has \(d\) attributes, variables or features.
  or
  symmetric \([1:n,1:n]\) distance matrix, e.g. `as.matrix(dist(Data, method))`

- **ClusterNo**
  A number \(k\) which defines \(k\) different clusters to be built by the algorithm.

- **Type**
  Type of Projection method, either
  - NerV [Venna et al., 2010]
  - Pswarm [Thrun/Ultsch, 2020b]
  - MDS [Torgerson, 1952]
  - Uwot [McInnes et al., 2018]
  - CCA [Demartines/Herault, 1995]
  - Sammon [Sammon, 1969]
  - t-SNE [Van der Maaten/Hinton, 2008]

- **StructureType**
  Either compact (TRUE) or connected (FALSE), see discussion in [Thrun, 2018]

- **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 

- **PlotTree**
  TRUE: Plots the dendrogram, FALSE: no plot

- **PlotMap**
  Plots the topographic map [Thrun et al., 2016].

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

**Details**

The first idea of using non-PCA projections for clustering was published by [Bock, 1987] as a definition. However, to the knowledge of the author, it was not applied to any data. The coexistence of projection and clustering was introduced in [Thrun/Ultsch, 2017].

Projection-based clustering is based on a nonlinear projection of high-dimensional data into a two-dimensional space [Thrun/Ultsch, 2020b]. Typical projection-methods like t-distributed stochastic neighbor embedding (t-SNE) [Van der Maaten/Hinton, 2008], or neighbor retrieval visualizer (NerV) [Venna et al., 2010] are used project data explicitly into two dimensions disregarding the subspaces of higher dimension than two and preserving only relevant neighborhoods in high-dimensional data. In the next step, the Delaunay graph [Delaunay, 1934] between the projected points is calculated, and each vertex between two projected points is weighted with the high-dimensional distance between the corresponding high-dimensional data points. Thereafter the shortest path between every pair of points is computed using the Dijkstra algorithm [Dijkstra, 1959]. The shortest paths are then used in the clustering process, which involves two choices depending on
the structure type in the high-dimensional data [Thrun/Ultsch, 2020b]. This Boolean choice can be
decided by looking at the topographic map of high-dimensional structures [Thrun/Ultsch, 2020a].
In a benchmarking of 34 comparable clustering methods, projection-based clustering was the only
algorithm that always was able to find the high-dimensional distance or density-based structure of
the dataset [Thrun/Ultsch, 2020b].

It should be noted that it is preferable to use a visualization for the Generalized U-Matrix like
the topographic map plotTopographicMap of [Thrun et al., 2016] to evaluate the choice of the
boolean parameter StructureType and the clustering, improve it or set the number of clusters
appropriately. A comparison with 32 clustering algorithms showed that PBC is always able to find
the correct cluster structure while the best of the 32 clustering algorithms varies depending on the
dataset [Thrun/Ultsch, 2020].

The first systematic comparison to other DR clustering methods like Projection-Pursuit Methods
ProjectionPursuitClustering, supspace clustering methods SubspaceClustering, and CA-
Based Clustering can be found in [Thrun/Ultsch, 2020a]. For PCA-based clustering meth-
ods please see TandemClustering.

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

[Bock, 1987] Bock, H.: On the interface between cluster analysis, principal component analy-
sis, and multidimensional scaling. Multivariate statistical modeling and data analysis, (pp. 17-34),

Federation of Classification Societies (IFCS). pp. 250-251, Tokai University, Japanese Classifica-
tion Society (JCS), Tokyo, Japan August 7-10, 2017.

Distance and Density based Clusters in High-Dimensional Data, Journal of Classification, in press,

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,

[McInnes et al., 2018] McInnes, L., Healy, J., & Melville, J.: Umap: Uniform manifold approxima-
Examples

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

---

**Description**

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

**Usage**

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

**Details**

- Size 1000, Dimensions 3, stored in `Chainlink$Data`
- Classes 2, stored in `Chainlink$Cls`

---


References


Examples

data(Chainlink)
str(Chainlink)

Clusterability MDplot  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 probability to possess cluster structures, if the first component of the PCA projection is multimodal [Adolfsson et al., 2019]. As the dip test is less exact than the MDplot [Thrun et al., 2020], p-values above 0.05 can be given for MD-plots which are clearly multimodal.

An alternative investigation of clusterability can be performed by inspecting the topographic map of the Generalized U-Matrix for a specific projection method using the ProjectionBasedClustering and GeneralizedUmatrix packages on CRAN, see [Thrun/Ultsch, 2021] for details.

Usage

ClusterabilityMDplot(DataOrDistance, Method, na.rm=FALSE, ...)

Arguments

DataOrDistance Either a dataset[1:n,1:d] of n cases and d features or a symmetric distance matrix [1:d,1:d] or multiple data sets or distances 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).
na.rm Statistical testing will not work with missing values, if TRUE values are imputed with averages
... Further arguments like main, and ordering
Details

Use the method of [Adolfsson et al., 2019] specified as pca plus dip-test (PCA dip) per default without scaling or standardization of data because this step should never be done automatically. In [Thrun, 2020] the standardization and scaling did not improve the results.

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 list.

Beware, as shown below, this test fails for almost touching clusters of Tetra and is difficult to interpret on WingNut but with overlayed with a robustly 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].

Based on [Thrun, 2020], the author of this function disagrees with [Adolfsson et al., 2019] as to the preference which clusterability method should be used because the approach "distance" is not preferable for density-based cluster structures.

Value

ggplot2 plotter handle

Note

"none" seems to call dip.test in clusterabilitytest with high-dimensional data. In that case dip.test just vectorizes the matrix of the data which does not make any sense. Since this could be a bug, the "none" option should not be used.

Imputation does not work for distance matrices. Imputation is still experimental. It is advised to impute missing values before using this function

Author(s)

Michael Thrun

References


ClusterApply

Applies a function over grouped data

Description

Applies a given function to each dimension d of data separately for each cluster

Usage

ClusterApply(DataOrDistances,FUN,Cls,...)
ClusterApply

Arguments

DataOrDistances

[1:n,1:d] with: if d=n and symmetric then distance matrix assumed, otherwise:
[1:n,1:d] matrix of defining the dataset that consists of n cases or d-dimensional
data points. Every case has d attributes, variables or features.

FUN

Function to be applied to each cluster of data and each column of data

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.

Details

Applies a given function to each feature of each cluster of data using the clustering stored in Cls
which is the cluster identifiers for all rows in data. If missing, all data are in first cluster. The
main output is FUNPerCluster[i] which is the result of FUN for the data points in cluster of
UniqueClusters[i] named with the function’s name used.

In case of a distance matrix an automatic classical multidimensional scaling transformation of dis-
tances to data is computed. Number of dimensions is selected by the minimal stress w.r.t. the
possible output dimensions of cmdscale.

If FUN has not function name, then ResultPerCluster is given back.

Value

UniqueClusters   The unique clusters in Cls
FUNPerCluster    a matrix of [1:k,1:d] of d features and k clusters

Author(s)

Felix Pape, Michael Thrun

Examples

## one dataset
data(Hepta)
Data=Hepta$Data
Cls=Hepta$Cls
# mean per cluster
ClusterApply(Data,mean,Cls)

# Mean per cluster of MDS transformation
# Beware, this is not the same!
ClusterApply(as.matrix(dist(Data)),mean,Cls)

## Not run:
Iris=datasets::iris
ClusterChallenge

Generates a Fundamental Clustering Challenge based on specific artificial datasets.

Description

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

Usage

ClusterChallenge(Name, SampleSize, PlotIt=FALSE, PointSize=1, Plotter3D="rgl", ...)

Arguments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>string, either 'Atom', 'Chainlink', 'EngyTime', 'GolfBall', 'Hepta', 'Lsun3D', 'Target', 'Tetra' 'TwoDiamonds' 'WingNut</td>
</tr>
<tr>
<td>SampleSize</td>
<td>Size of Sample higher than 300, preferable above 500</td>
</tr>
<tr>
<td>PlotIt</td>
<td>TRUE: Plots the challenge with ClusterPlotMDS</td>
</tr>
<tr>
<td>PointSize</td>
<td>If PlotIt=TRUE: see ClusterPlotMDS</td>
</tr>
<tr>
<td>Plotter3D</td>
<td>If PlotIt=TRUE: see ClusterPlotMDS</td>
</tr>
<tr>
<td>...</td>
<td>If PlotIt=TRUE: further arguments for ClusterPlotMDS</td>
</tr>
</tbody>
</table>
Details

A detailed description of the datasets can be found in [Thrun/Ultsch 2020]. 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


See Also

ClusterPlotMDS

Examples

## Not run:
ClusterChallenge("Chainlink",2000,PlotIt=TRUE)

## End(Not run)

------------------
ClusterCount ClusterCount
------------------

Description

Calculates statistics for clustering in each group of the data points

Usage

ClusterCount(Cls,Ordered=FALSE)
ClusterCreateClassification

Arguments

Cls 1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.

Ordered Optional, boolean, if TRUE: the output is ordered increasingly by cluster labels in UniqueClusters.

Details

The ordering of the output is defined by the first occurrence of every cluster label in Cls in the default setting (Ordered=FALSE). If non-finite values are given in the numerical vector, they are set to the "9999" cluster.

Value

UniqueClusters The unique clusters in Cls
CountPerCluster The number of data points in the corresponding unique clusters.
NumberOfClusters The number of clusters
ClusterPercentages The percentages of datapoints belonging to a cluster for each cluster

Author(s)

Michael Thrun

Examples

data('Hepta')
Cls=Hepta$Cls
ClusterCount(Cls)

Description

Creates a Cls from arbitrary list of objects

Usage

ClusterCreateClassification(Objects)

Arguments

Objects Listed objects, for example factor
ClusterDaviesBouldinIndex

Details

See example

Value

LIST, with

Cls [1:n] numerical vector with n numbers defining the labels of the classification. It has 1 to k unique numbers representing the arbitrary labels of the classification.

ClusterNames ClusterNames defined which names belong to which unique number

Author(s)

Michael Thrun

Examples

```r
## Not run:
Iris=datasets::iris
SomeFactors=Iris$Species
V=ClusterCreateClassification(SomeFactors)
Cls=V$Cls
V$ClusterNames
table(Cls,SomeFactors)
## End(Not run)
```

ClusterDaviesBouldinIndex

Davies Bouldin Index

Description

Internal (i.e. without prior classification) cluster quality measure called Davies Bouldin index for a given clustering published in [Davies/Bouldin, 1979].

Usage

ClusterDaviesBouldinIndex(Cls, Data,...)

Arguments

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

Data matrix, [1:d,1:n] dataset of d variables and n cases

... Further arguments passed on to the index.DB function of clusterSim
ClusterDendrogram

Details

Wrapper for index.DB. Davies Bouldin index is defined in [Davies/Bouldin, 1979]. Best clustering scheme essentially minimizes the Davies-Bouldin index because it is defined as the function of the ratio of the within cluster scatter, to the between cluster separation.[Davies/Bouldin, 1979].

Value

List of
DaviesBouldinIndex
  scalar, Davies Bouldin index
Object further information stored in index.DB

Author(s)

Michael Thrun

References


Examples

data("Hepta")
Cls=kmeansClustering(Hepta$Data, ClusterNo = 7, Type="Hartigan")$Cls
ClusterDaviesBouldinIndex(Cls, Hepta$Data)[1]

data("Hepta")
ClsWellSeperated=kmeansClustering(Hepta$Data, ClusterNo = 7, Type="Steinley")$Cls
ClusterDaviesBouldinIndex(ClsWellSeperated, Hepta$Data)[1]

ClusterDendrogram  Cluster Dendrogram

Description

Presents a dendrogram of a given tree using a color sequence for the branches defined from the highest cluster size to the lowest cluster size.

Usage

ClusterDendrogram(TreeOrDendrogram, ClusterNo,
Colorsequence, main='Name of Algorithm')
ClusterDendrogram

Arguments

**TreeOrDendrogram**

Either object of hclust defining the tree, third list element of hierarchical cluster algorithms of this package

or

Object of class dendrogram, second list element of hierarchical cluster algorithms.

**ClusterNo**

k number of clusters for cutree.

**Colorsequence**

[1:k] character vector of colors, per default the colorsquence defined in the

*DataVisualizations* is used

**main**

Title of plot

Details

Requires the package *dendextend* to work correctly.

Value

In mode invisible:

[1:n] numerical vector defining the clustering of k clusters; this classification is the main output of the algorithm.

Author(s)

Michael Thrun

See Also

*cutree*, *hclust*

Examples

data(Lsun3D)
listofh=HierarchicalClustering(Lsun3D$Data,0,'SingleL')
Tree=listofh$Object
#given colors are per default:
#"magenta" "yellow" "black" "red"
ClusterDendrogram(Tree, 4,main='Single Linkage Clustering')

listofh=HierarchicalClustering(Lsun3D$Data,4)
ClusterCount(listofh$Cls)
#c1 is magenta, c2 is red, c3 is yellow, c4 is black
#because the order of the cluster sizes is
#c1,c3,c4,c2
Description

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

Usage

ClusterDistances(FullDistanceMatrix, Cls,
Names, PlotIt = FALSE)

ClusterIntraDistances(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. Details and definitions can be found in [Thrun, 2021].

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 length as the vector of the upper triangle of the complete distance matrix.

Author(s)

Michael Thrun

References

ClusterDunnIndex

See Also

MDplot
ClusterInterDistances

Examples

data(Hepta)
Distance=as.matrix(dist(Hepta$Data))

interdists=ClusterDistances(Distance,Hepta$Cls)

Description

Internal (i.e. without prior classification) cluster quality measure called Dunn index for a given clustering published in [Dunn, 1974].

Usage

ClusterDunnIndex(Cls,DataOrDistances,

DistanceMethod="euclidean",Silent=TRUE,Force=FALSE,...)

Arguments

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

DataOrDistances matrix, DataOrDistance[1:n,1:n] symmetric matrix of dissimilarities, if variable unsymmetric DataOrDistance[1:d,1:n] is assumed as a dataset and the euclidean distances are calculated of d variables and n cases

DistanceMethod Optional, one of 39 distance methods of parDist of package parallelDist, if Data matrix is chosen above

Silent TRUE: Warnings are shown

Force TRUE: force computing in case of numerical instability

Details

Dunn index is defined as Dunn=min(InterDist)/max(IntraDist). Well seperated clusters have usually a dunn index above 1, for details please see [Dunn, 1974].
ClusterEqualWeighting

Value

List of

- **Dunn**: scalar, Dunn Index
- **IntraDist**: [1:k] numerical vector of minimal intra cluster distances per given cluster
- **InterDist**: [1:k] numerical vector of minimal inter cluster distances per given cluster

Author(s)

Michael Thrun

References


Examples

data("Hepta")
Cls=kmeansClustering(Hepta$Data,ClusterNo = 7,Type="Hartigan")$Cls
ClusterDunnIndex(Cls,Hepta$Data)

data("Hepta")
ClsWellSeperated=kmeansClustering(Hepta$Data,ClusterNo = 7,Type="Steinley")$Cls
ClusterDunnIndex(ClsWellSeperated,Hepta$Data)

ClusterEqualWeighting

Description

Weights clusters equally

Usage

ClusterEqualWeighting(Cls, Data, MinClusterSize)

Arguments

- **Cls**: 1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.
- **Data**: Optional, [1:n,1:d] matrix of dataset consisting of n cases of d-dimensional data points. Every case has d attributes, variables or features.
- **MinClusterSize**: Optional, scalar defining the number of cases m that each cluster should have.
Details

Balance clusters such that their sizes are the same by subsampling the larger cluster. If MinClusterSize is missing the number of cases per cluster is set to the smallest cluster size. For clusters sizes smaller than MinClusterSize, sampling with replacement is turned on, i.e. up sampling. For clusters sizes equal to MinClusterSize, no sampling is performed.

Value

List of

BalancedCls Vector of Cls such that all clusters have the same sizes spezified by MinClusterSize
BalancedInd index such that BalancedCls = Cls[BalancedInd]
BalancedData NULL if missing, otherwise, Data[BalancedInd,]

Author(s)

Alfred Ultsch (matlab), reimplemented by Michael Thrun

Examples

data(Hepta)
ClusterEqualWeighting(Hepta$Cls,Hepta$Data,5)

Description

ClusterAccuracy

Usage

ClusterAccuracy(PriorCls,CurrentCls,K=9)

Arguments

PriorCls Ground truth,[1:n] numerical vector with n numbers defining the classification. It has k unique numbers representing the arbitrary labels of the clustering.
CurrentCls Main output of the clustering, [1:n] numerical vector with n numbers defining the classification. It has k unique numbers representing the arbitrary labels of the clustering.
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 [Thrun et al., 2018]. Beware that in contrast to ClusterMCC, the labels can be arbitrary. However, accuracy is only a valid quality measure if the clusters are balanced (of nearly equal size). Otherwise please use ClusterMCC.

In contrast to the F-measure, "Accuracy tends to be naturally unbiased, because it can be expressed in terms of a binomial distribution: A success in the underlying Bernoulli trial would be defined as sampling an example for which a classifier under consideration makes the right prediction. By definition, the success probability is identical to the accuracy of the classifier. The i.i.d. assumption implies that each example of the test set is sampled independently, so the expected fraction of correctly classified samples is identical to the probability of seeing a success above. Averaging over multiple folds is identical to increasing the number of repetitions of the Binomial trial. This does not affect the posterior distribution of accuracy if the test sets are of equal size, or if we weight each estimate by the size of each test set." [Forman/Scholz, 2010]

Value

Single scalar of Accuracy between zero and one

Author(s)

Michael Thrun

References

[Thrun et al., 2018] 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


See Also

ClusterMCC

Examples

# Influence of random sets/ random starts on k-means
data('Hepta')
Cls=kmeansClustering(Hepta$Data,7,Type = "Hartigan",nstart=1)
table(Cls$Cls,Hepta$Cls)
ClusterAccuracy(Hepta$Cls,Cls$Cls)
ClusterInterDistances

Computes Inter-Cluster Distances

Description

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

Usage

ClusterInterDistances(FullDistanceMatrix, Cls, Names, PlotIt=FALSE)

Arguments

FullDistanceMatrix
[1:n,1:n] symmetric distance matrix

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

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. Details and definitons can be found in [Thrun, 2021].

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 length as the vector of the upper triangle of the complete distance matrix.

Author(s)

Michael Thrun

References

See Also

MDplot
ClusterDistances

Examples

data(Hepta)
Distance=as.matrix(dist(Hepta$Data))

interdists=ClusterInterDistances(Distance,Hepta$Cls)

ClusterMCC

Matthews Correlation Coefficient (MCC)

Description

Matthews correlation coefficient generalized to the multiclass case (a.k.a. R_K statistic).

Usage

ClusterMCC(PriorCls, CurrentCls)

Arguments

PriorCls Ground truth,[1:n] numerical vector with n numbers defining the classification. It has k unique numbers representing the labels of the clustering.

CurrentCls Main output of the clustering, [1:n] numerical vector with n numbers defining the classification. It has k unique numbers representing the labels of the clustering.

Details

Contrary to accuracy, the MCC is balanced measure which can be used even if the classes are of very different sizes. When there are more than two labels the MCC will no longer range between -1 and +1. Instead the minimum value will be between -1 and 0 depending on the true distribution. The maximum value is always +1. Beware that in contrast to ClusterAccuracy, the labels cannot be arbitrary. Instead each label of PriorCls and CurrentCls has to be mapped to the same cluster of data points. Typically this has to be ensured manually.

Value

Single scalar of MCC in a range described in details.

Note

If No. of Clusters is not equivalent, internally the number is allgined with zero datapoints belonging to the missing clusters.
ClusterNoEstimation

Author(s)
Michael Thrun

References

See Also
ClusterAccuracy

Examples
#Beware that algorithm arbitrary defines the labels
data(Hepta)
V=kmeansClustering(Hepta$Data,Type = "Hartigan",7)
table(V$,Hepta$Cls)
#result is only valid if the above issue is resolved manually
ClusterMCC(Hepta$Cls,V$,Cls)

ClusterNoEstimation 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 clusters 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 specific projection method using the ProjectionBaseClustering and GeneralizedUmatrix packages on CRAN, see [Thrun/Ultsch, 2021] for details.

Usage
ClusterNoEstimation(DataOrDistances, ClsMatrix = NULL, MaxClusterNo,
ClusterIndex = "all", Method = NULL, MinClusterNo = 2,
Silent = TRUE,PlotIt=FALSE,SelectByABC=TRUE,Colorsequence,...)
Arguments

DataOrDistances
Either \([1:n,1:d]\) matrix of dataset to be clustered. It consists of \(n\) cases of \(d\)-dimensional data points. Every case has \(d\) attributes, variables or features.
or
Symmetric \([1:n,1:n]\) distance matrix

ClsMatrix
\([1:n,1:(\text{MaxClusterNo})]\) matrix of clusterings each columns is defined as:
\(1:n\) numerical vector of numbers defining the classification as the main output of the clustering algorithm for the \(n\) cases of data. It has \(k\) unique numbers representing the arbitrary labels of the clustering.
(see also details (2) and (3)), must be specified if method = NULL

MaxClusterNo
Highest number of clusters 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:

ClusterIndex
String or vector of strings with the indicators to be calculated (see details (1)), default = "all"

MinClusterNo
Lowest number of clusters to be checked, default = 2

Silent
If TRUE status messages are output, default = FALSE

PlotIt
If TRUE plots fanplot with proposed cluster numbers

SelectByABC
If PlotIt=TRUE, TRUE: Plots group A of ABCanalysis of the most important ones (highest overlap in indicators), FALSE: plots all indicators

Colorsequence
Optional, character vector of sufficient length of colors for the fan plot. If the sequence is too long the first part of the sequence is used.

Details

Each column of ClsMatrix has to have at least two unique clusters defined. Otherwise the function will stop.

(1)
These can be specified individually or as a vector via the parameter index. If you enter ’all’, all key figures are calculated.

(2)
The indicators kl, duda, pseudot2, beale, frey and mcclain require a clustering for MaxClusterNo+1 clusters. If these key figures are to be calculated, this clustering must be specified in cls.

(3)
The indicator kl requires a clustering for MinClusterNo-1 clusters. If this key figure is to be calculated, this clustering must also be specified in cls. For the case MinClusterNo = 2 no clustering for 1 has to be given.
The following methods can be used to create clusterings:
"kmeans," "DBSclustering", "DivisiveAnalysisClustering", "FannyClustering", "ModelBasedClustering", "SpectralClustering" or all methods found in HierarchicalClustering.

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

If a distances matrix is given, then ProjectionBasedClustering is required to be accessible.

Value

<table>
<thead>
<tr>
<th>Indicators</th>
<th>A table of the calculated indicators except Duda, Pseudot2 and Beale</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterNo</td>
<td>The recommended number of clusters for each calculated indicator</td>
</tr>
<tr>
<td>ClsMatrix</td>
<td>[1:n,MinClusterNo:(MaxClusterNo)] Output of the clusterings used for the calculation</td>
</tr>
<tr>
<td>HierarchicalIndicators</td>
<td>Either NULL or the values for the indicators Duda, Pseudot2 and Beale in case of hierarchical cluster procedures, if calculated</td>
</tr>
</tbody>
</table>

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. Colorsequence works if DataVisualizations 1.1.13 is installed (currently only on github available).

Author(s)

Peter Nahrgang, revised by Michael Thrun (2021)

References


Examples

# Reading the iris dataset from the standard R-Package datasets
data <- as.matrix(iris[,1:4])
MaxClusterNo = 7

# Creating the clusterings for the data set
#(here with method complete) for the number of clusters 2 to 8
hc <- hclust(dist(data), method = "complete")
clsm <- matrix(data = 0, nrow = dim(data)[1],
...
ClusterNormalize

Cluster Normalize

Description

Values in Cls are consistently recoded to positive consecutive integers

Usage

ClusterNormalize(Cls)

Arguments

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

Details

For recoding depending on cluster size please see ClusterRenameDescendingSize.

Value

The renamed classification. A vector of clusters recoded to positive consecutive integers.

Author(s)

.

See Also

ClusterRenameDescendingSize
Examples

data('Lsun3D')
Cls=Lsun3D$Cls
# not descending cluster numbers
Cls[Cls==1]=543
Cls[Cls==4]=1

# Now ordered consecutively
ClusterNormalize(Cls)

ClusterPlotMDS

Plot Clustering using Dimensionality Reduction by MDS

Description

This function uses a projection method to perform dimensionality reduction (DR) on order to visualize the data as 3D data points colored by a clustering.

Usage

ClusterPlotMDS(DataOrDistances, Cls, main = "Clustering",
DistanceMethod = "euclidean", OutputDimension = 3,
PointSize=1,Plotter3D="rgl",Colorsequence, ...)

Arguments

DataOrDistances Either nonsymmetric [1:n,1:d] datamatrix of n cases and d features or symmetric [1:n,1:n] distance matrix
Cls 1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.
main String, title of plot
DistanceMethod Method to compute distances, default "euclidean"
OutputDimension Either two or three depending on user choice
PointSize Scalar defining the size of points
Plotter3D In case of 3 dimensions, choose either "plotly" or "rgl",
Colorsequence [1:k] character vector of colors, per default the colorsquence defined in the DataVisualizations is used
...

Please see Plot3D in DataVisualizations
Details

If dataset has more than 3 dimesions, mds is performed as defined in the smacof [De Leeuw/Mair, 2011]. If smacof package is not installed, classical metric MDS (see Def. in [Thrun, 2018]) is performed. In both cases, the first OutputDimension are visualized. Points are colored by the labels (Cls).

In the special case that the dataset has not more than 3 dimensions, all dimensions are visualized and no DR is performed.

Value

The rgl or plotly plot handler depending on Plotter3D

Note

If DataVisualizations is not installed a 2D plot using native plot function is shown.

If MASS is not installed, classical metric MDS is used, see [Thrun, 2018] for definition.

Author(s)

Michael Thrun

References


See Also

Plot3D

Examples

data(Hepta)
ClusterPlotMDS(Hepta$Data,Hepta$Cls)

data(Leukemia)
ClusterPlotMDS(Leukemia$DistanceMatrix,Leukemia$Cls)
Description

Redefines some or all Clusters of Clustering such that the names of the numerical vectors are defined by

Usage

ClusterRedefine(Cls, NewLabels, OldLabels)

Arguments

Cls 1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.

NewLabels [1:p], p<=k labels (identifiers) of clusters to be changed with

OldLabels Optional, [1:p], p<=k labels (identifiers) of clusters to be changed, default [1:k] unique cluster Ids of Cls

Details

The same ordering of NewLabels and OldLabels is assumed, i.e., the mapping is defined by OldLabels[i] -> NewLabels[i] with i in [1:p]. NewLabels can also be a vector for strings, for example for plotting.

Value

Cls[1:n] numerical vector named after the row names of data

Author(s)

Michael Thrun

Examples

data('Lsun3D')
Cls=Lsun3D$Cls
Data=Lsun3D$Data#
#prior
ClsNew=unique(Cls)+10
#Redefined Clustering
NewCls=ClusterRedefine(Cls,ClsNew)

table(Cls,NewCls)

#require(DataVisualizations)
ClusterRename

Renames Clustering such that the names of the numerical vectors are the row names of DataOrDistances.

Usage

ClusterRename(Cls, DataOrDistances)

Arguments

Cls 1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.

DataOrDistances Either nonsymmetric [1:n,1:d] datamatrix of n cases and d features or symmetric [1:n,1:n] distance matrix

Details

If DataOrDistances is missing or if inconsistent length, nothing is done.

Value

Cls[1:n] numerical vector named after the row names of data

Author(s)

Michael Thrun

Examples

data('Hepta')
Cls=Hepta$Cls
Data=Hepta$Data#
#prior
Cls
#Named Clustering
ClusterRename(Cls,Data)
Cluster Rename Descending Size

Description

Renames the clusters of a classification in descending order.

Usage

ClusterRenameDescendingSize(Cls,

ProvideClusterNames=FALSE)

Arguments

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

ProvideClusterNames TRUE: Provides in separate output new and old k numbers, FALSE: simple output

Details

Beware: output changes in this function depending on ProvideClusterNames in order to be congruent to prior code in a large variety of other packages.

Value

ProvideClusterNames==FALSE:

RenamedCls The renamed classification. A vector of clusters, were the largest cluster is C1 and so forth

ProvideClusterNames==TRUE: List V with

RenamedCls The renamed classification. A vector of clusters, were the largest cluster is C1 and so forth

ClusterName [1:k,1:2] matrix of k new numbers and prior numbers

Author(s)

Michael Thrun, Alfred Ultsch

See Also

ClusterNormalize
Examples

data('Lsun3D')
Cls=Lsun3D$Cls
# not descending cluster numbers
Cls[Cls==1]=543
Cls[Cls==4]=1

# Now ordered per cluster size and descending
ClusterRenameDescendingSize(Cls)

Description

Shannon Information [Shannon, 1948] for each column in ClsMatrix.

Usage

ClusterShannonInfo(ClsMatrix)

Arguments

ClsMatrix [1:n,1:C] matrix of C clusterings each column is defined as:
1:n numerical vector of numbers defining the classification as the main output
of the clustering algorithm for the n cases of data. It has k unique numbers
representing the arbitrary labels of the clustering.

Details

Info[,d] = sum(-p * log(p)/MaxInfo) for all unique cases with probability p in ClsMatrix[,c] for a
column with k clusters MaxInfo = -(1/k)*log(1/k)

Value

Info [1:max.nc,1:C] matrix of Shannon information as defined in details, each column
represents one Cls of ClsMatrix, each row yields the information of one cluster
up to the ClusterNo k, if k<max.nc (highest number of clusters) then NaN are
filled.

ClusterNo Number of Clusters k found for each Cls respectively
MaxInfo max per column of Info
MinInfo min per column of Info
MedianInfo median per column of Info
MeanInfo mean per column of Info
ClusterUpsamplingMinority

Note

reeimplemented from Alfred's Ultsch Matlab version but not verified yet.

Author(s)

Michael Thrun

References


Examples

# Reading the iris dataset from the standard R-Package datasets
data <- as.matrix(iris[,1:4])
max.nc = 7
# 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 = max.nc)
for (i in 2:(max.nc+1)) {
  clsm[,i-1] <- cutree(hc,i)
}
ClusterShannonInfo(clsm)

ClusterUpsamplingMinority

Cluster Up Sampling using SMOTE for minority cluster

Description

Wrapper for one specific internal function of L. Torgo who implemented there the relevant part of the SMOTE algorithm [Chawla et al., 2002].

Usage

ClusterUpsamplingMinority(Cls, Data, MinorityCluster,

Percentage = 200, knn = 5, PlotIt = FALSE)
ClusterUpsamplingMinority

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cls</td>
<td>1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.</td>
</tr>
<tr>
<td>Data</td>
<td>[1:n,1:d] datamatrix of n cases and d features</td>
</tr>
<tr>
<td>MinorityCluster</td>
<td>scalar defining the number of the cluster to be upsampled</td>
</tr>
<tr>
<td>Percentage</td>
<td>percentage above 100 of who many samples should be taken</td>
</tr>
<tr>
<td>knn</td>
<td>k nearest neighbors of SMOTE algorithm</td>
</tr>
<tr>
<td>PlotIt</td>
<td>TRUE: plots the result using ClusterPlotMDS</td>
</tr>
</tbody>
</table>

Details

the number of items m is defined by the scalar Percentage and the up sampling is combined with the Data and the Cls to DataExt and ClsExt such that the sample is placed thereafter.

Value

List with

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClsExt</td>
<td>1:(n+m) numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.</td>
</tr>
<tr>
<td>DataExt</td>
<td>[1:(n+m),1:d] datamatrix of n cases and d features</td>
</tr>
</tbody>
</table>

Author(s)

L. Torgo

References


Examples

```
data(Lsun3D)
Data=lsun3D$Data
Cls=lsun3D$Cls
table(Cls)

V=ClusterUpsamplingMinority(Cls,Data,4,1000)
table(V$ClsExt)
```
Description
Density-Based Spatial Clustering of Applications with Noise of [Ester et al., 1996].

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 of 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 NULL, automatic estimation is performed 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]. If NULL, 2.5 percent of points is selected.

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


Examples

data('Hepta')

out = DBSCAN(Hepta$Data, Radius=NULL, minPts=NULL, PlotIt=FALSE)

## Not run:
# 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 = FALSE,
    UpperLimitRadius = parameters$Radius
  )$Cls
  if (length(unique(Cls9)) < 5)
    BestAcc[i] = DatabionicSwarm::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,
  UpperLimitRadius = parameters$Radius,
  PlotIt = TRUE
)
DBSclusteringAndVisualization

Databionic Swarm (DBS) Clustering and Visualization

Description

Swarm-based clustering by exploiting self-organization, emergence, swarm intelligence and game theory published in [Thrun/Ultsch, 2021].

Usage

DatabionicSwarmClustering(DataOrDistances, ClusterNo = 0,
StructureType = TRUE, DistancesMethod = NULL,
PlotTree = FALSE, PlotMap = FALSE, PlotIt = FALSE, Data)

Arguments

DataOrDistances

Either nonsymmetric [1:n,1:d] numerical matrix of a dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

or

symmetric [1:n,1:n] distance matrix, e.g. as.matrix(dist(Data,method))

ClusterNo

Number of Clusters, if zero a the topographic map is plotted. 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 if GeneralizedUmatrix is installed.

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.
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 for evaluation purposes), then please use the **DatabionicSwarm** package directly and read the vignette there. Databionic swarm is like k-means a stochastic algorithm meaning that the clustering and visualization may change between trials.

Value
List of

- **Cls**  1:n numerical vector of numbers defining the classification as the main output of the clustering algorithm for the n cases of data. It has k unique numbers representing the arbitrary labels of the clustering.

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 it takes longer than a day for a result.

Author(s)
Michael Thrun

References


See Also
**Pswarm, DBSclustering, GeneratePswarmVisualization**

Examples
```r
# Generate random but small non-structured data set
data = cbind(          
sample(1:100, 300, replace = TRUE),
    sample(1:100, 300, replace = TRUE),
```
sample(1:100, 300, replace = TRUE)
)
# Make sure there are no structures
# (sample size is small and still could generate structures randomly)
if(requireNamespace('DatabionicSwarm', quietly = TRUE)){
  Data = DatabionicSwarm::RobustNormalization(data, Centered = TRUE)
  #DataVisualizations::Plot3D(Data)

  # No structures are visible
  # Topographic map looks like "egg carton"
  # with every point in its own valley
 Cls = DatabionicSwarmClustering(Data, 0, PlotMap = TRUE)
} else{
  # only for testing purposes of CRAN!
  # in case CRAN tests with no suggest packages available
  # please use always some kind of standardization!
 Cls = DatabionicSwarmClustering(data, 0, PlotMap = TRUE)
}

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

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

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

# entangled, complex, and non-linear separable structures
# Not run:
# takes too long for CRAN tests

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

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

# Experiment with parameter StructureType only
# reveals that clustering is appropriate
# if StructureType=FALSE
Cls = DatabionicSwarmClustering(Chainlink$Data,
  2,
  StructureType = FALSE,
  PlotMap = TRUE)

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

Density Peak Clustering algorithm using the Decision Graph

Description


Usage

DensityPeakClustering(DataOrDistances, Rho, Delta, Dc, Knn = 7, DistanceMethod = "euclidean", PlotIt = FALSE, Data, ...)

Arguments

DataOrDistances
Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] non symmetric data matrix of n cases and d variables

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

DistanceMethod
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.

Details

The densityClust algorithm does not decide the k number of clusters, this has to be done by the parameter setting. This is contrary to the other version of the algorithm from 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).
DensityPeakClustering

Value

If Rho and Delta are set:

list of

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

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


See Also

ADPclustering
densityClust

denisty

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
Divisive Analysis Clustering (diana) of [Rousseeuw/Kaufman, 1990, pp. 253-279]

Usage

```r
DivisiveAnalysisClustering(DataOrDistances, ClusterNo, PlotIt=FALSE, Standardization=TRUE, PlotTree=FALSE, Data,...)
```

Arguments

- **DataOrDistances**
  - `[1:n,1:d]` matrix of dataset to be clustered. It consists of `n` cases of `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` and `PlotTree=TRUE`, 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.

- **PlotTree**
  - TRUE: Plots the dendrogram, FALSE: no plot.

- **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

- **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 hierarchical clustering algorithm.

- **Object**
  - Object defined by clustering algorithm as the other output of this algorithm.
EngyTime

Author(s)
Michael Thrun

References

Examples
data('Hepta')
CA=DivisiveAnalysisClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

print(CA$Object)
plot(CA$Object)
ClusterDendrogram(CA$Dendrogram,7,main='DIANA')

EngyTime introduced in [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

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 given ranges of impact factors sigma.

**Usage**

```r
EntropyOfDataField(Data,

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

   , PlotIt = FALSE)
```

**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 curve 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 define the correct data field. It follows, that the influence radius is $3/\sqrt{2}\times\sigma$ (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**

EstimateRadiusByDistance

Examples

```r
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)])
```

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

```r
EstimateRadiusByDistance(DistanceMatrix)
```

Arguments

- **DistanceMatrix** 
  [1:n,1:n] symmetric distance Matrix of n cases

Details

For density-based clustering algorithms like DBSCAN it is not always useful.

Value

Numerical scalar defining the radius

Note

Symmetric matrix is assumed.

Author(s)

Michael Thrun

References


See Also

GeneratePmatrix
FannyClustering

**Examples**

data('Hepta')
DistanceMatrix=as.matrix(dist(Hepta$Data))
Radius=EstimateRadiusByDistance(DistanceMatrix)

**Description**
...

**Usage**
FannyClustering(DataOrDistances,ClusterNo,
PlotIt=FALSE,Standardization=TRUE,...)

**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.

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 second output of this algorithm
GenieClustering

Author(s)
Michael Thrun

References

Examples

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

GenieClustering

Genie Clustering by Gini Index

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

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataOrDistances</td>
<td>[1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features. Alternatively, symmetric [1:n,1:n] distance matrix</td>
</tr>
<tr>
<td>ClusterNo</td>
<td>A number k which defines k different clusters to be build by the algorithm.</td>
</tr>
<tr>
<td>DistanceMethod</td>
<td>See <code>parDist</code>, for example 'euclidean', 'mahalanobis', 'manhattan' (cityblock), 'Jaccard', 'binary', 'canberra', 'maximum'. Any unambiguous substring can be given.</td>
</tr>
<tr>
<td>ColorTreshold</td>
<td>Draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given</td>
</tr>
<tr>
<td>...</td>
<td>further argument to <code>genie</code> like: thresholdGini Single numeric value in [0,1], threshold for the Gini index, 1 gives the standard single linkage algorithm</td>
</tr>
</tbody>
</table>

Details
Wrapper for Genie algorithm.
Value

List of

Cls If, ClusterNo>0: [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. Otherwise for ClusterNo=0: NULL

Dendrogram Dendrogram of hierarchical clustering algorithm

Object Ultrametric tree of hierarchical clustering algorithm

Author(s)

Michael Thrun

References


See Also

HierarchicalClustering

Examples

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

GolfBall introduced in [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
HCLclustering

References


Examples

data(GolfBall)
str(GolfBall)

HCLclustering

On-line Update (Hard Competitive learning) method

Description

Hard Competitive learning clustering published by [Ripley, 2007].

Usage

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

Arguments

Data
[1:n,1:d] matrix of dataset to be clustered. It consists of n cases of 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
HDDClustering

References


Examples

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

HDD clustering maximises the BIC criterion for a range of possible number of cluster up to ClusterNo. Per default the most general model is used, alternatively the parameter model="ALL" can be used to evaluate all possible models with BIC [Berge et al., 2012]. If specific properties of Data are known priorly please see hddc for specific model selection.
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)

Quirin Stier

References


Examples

# Hepta
data("Hepta")
Data = Hepta$Data
#Non-default parameter model
#can be set to evaluate all possible models
V = HDDClustering(Data=Data,ClusterNo=7,model="ALL")
Cls = V$Cls
ClusterAccuracy(Hepta$Cls, Cls)

# Not run:
library(HDclassif)
data(Crabs)
Data = Crabs[,,-1]
V = HDDClustering(Data=Data,ClusterNo=4,com_dim=1)

# End(Not run)

Hepta introduced in [Ulltisch, 2003]

Description

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

Usage

data("Hepta")

Details

Size 212, Dimensions 3, stored in Hepta$Data

Classes 7, stored in Hepta$Cls

References


Examples

data(Hepta)

str(Hepta)

HierarchicalClusterData

Internal function of Hierarchical Clusterer of Data

Description

Please use HierarchicalClustering. Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it. Uses stats package function 'hclust'.

Usage

HierarchicalClusterData(Data,ClusterNo=0,

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

ColorThreshold=0,Fast=FALSE,Cls=NULL,...)

Arguments

<table>
<thead>
<tr>
<th>Data</th>
<th>[1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterNo</td>
<td>A number k which defines k different clusters to be build by the algorithm.</td>
</tr>
<tr>
<td>Type</td>
<td>Methode der Clusterung: &quot;ward.D&quot;, &quot;ward.D2&quot;, &quot;single&quot;, &quot;complete&quot;, &quot;average&quot;, &quot;mcquitty&quot;, &quot;median&quot; or &quot;centroid&quot;.</td>
</tr>
</tbody>
</table>
HierarchicalClusterDists

DistanceMethod see `parDist`, for example 'euclidean', 'mahalanobis', 'manhatten' (cityblock), 'jaccard', 'binary', 'canberra', 'maximum'. Any unambiguous substring can be given.

ColorTreshold Draws cutline w.r.t. dendrogram 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

... In case of plotting further argument for plot, see `as.dendrogram`

Value

List of

Cls If, ClusterNo>0: [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. Otherwise for ClusterNo=0: NULL

Dendrogram Dendrogram of hierarchical clustering algorithm

Object Ultrametric tree of hierarchical clustering algorithm

Author(s)

Michael Thrun

See Also

HierarchicalClusterData
HierarchicalClusterDists
HierarchicalClustering

Examples

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

Description

Please use `HierarchicalClustering`. Cluster analysis on a set of dissimilarities and methods for analyzing it. Uses stats package function 'hclust'.
HierarchicalClusterDists

Usage

HierarchicalClusterDists(pDist, ClusterNo=0, Type="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 built by the algorithm.
Type 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
... In case of plotting further argument for plot, see as.dendrogram

Value

List of
Cls If, ClusterNo>0: [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. Otherwise for ClusterNo=0: NULL
Dendrogram Dendrogram of hierarchical clustering algorithm
Object Ultrametric tree of hierarchical clustering algorithm

Author(s)

Michael Thrun

See Also

HierarchicalClusterData
HierarchicalClusterDists
HierarchicalClustering

Examples

data('Hepta')
#out=HierarchicalClusterDists(as.matrix(dist(Hepta$Data)),ClusterNo=7)
HierarchicalClustering

Hierarchical Clustering

Description

Wrapper for various agglomerative hierarchical clustering algorithms.

Usage

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

Arguments

DataOrDistances Either nonsymmetric [1:n,1:d] numerical matrix of a dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

or

symmetric [1:n,1:n] distance matrix, e.g. as.matrix(dist(Data, method))

ClusterNo A number k which defines k different clusters to be built by the algorithm.

Type Method of cluster analysis: "Ward", "SingleL", "CompleteL", "AverageL" (UPGMA), "WPGMA" (mcquitty), "MedianL" (WPGMC), "CentroidL" (UPGMC), "Minimax", "MinEnergy", "Gini", "HDBSCAN", or "Sparse"

Fast If TRUE and fastcluster installed, then a faster implementation of the methods above can be used except for "Minimax", "MinEnergy", "Gini" or "HDBSCAN"

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"), HierarchicalDBSCAN (for HDBSCAN) or SparseClustering (for Sparse).

Details

Please see HierarchicalClusterData and HierarchicalClusterDists or the other functions listed above.

It should be noted that in case of "HDBSCAN" the number of clusters is manually selected by cutree to have the same convention as the other algorithms. Usually, "HDBSCAN" selects the number of clusters automatically.

Value

List of
HierarchicalDBSCAN

| Cls | If, ClusterNo>0: [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. Otherwise for ClusterNo=0: NULL |
| Dendrogram | Dendrogram of hierarchical clustering algorithm |
| Object | Ultrametric tree of hierarchical clustering algorithm |

**Author(s)**

Michael Thrun

**See Also**

HierarchicalClusterData, HierarchicalClusterDists, MinimalEnergyClustering.

**Examples**

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

**Description**

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

**Usage**

HierarchicalDBSCAN(DataOrDistances,minPts=4,

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

**Arguments**

- **DataOrDistances**
  - Either a [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features. or a [1:n,1:n] symmetric distance matrix.

- **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.
HierarchicalDBSCAN

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.
- Dendrogram Dendrogram of hierarchical clustering algorithm
- Tree Ultrametric tree of hierarchical clustering algorithm
- Object Object defined by clustering algorithm as the other output of this algorithm

Author(s)

Michael Thrun

References


Examples

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

data('Leukemia')
set.seed(1234)
CA=HierarchicalDBSCAN(Leukemia$DistanceMatrix)
#ClusterCount(CA$Cls)
#ClusterDendrogram(CA$Dendrogram,5,main='H-DBscan')
kmeansClustering  

K-Means Clustering

Description

Perform k-means clustering on a data matrix.

Usage

kmeansClustering(DataOrDistances, ClusterNo,  
Type = 'LBG', RandomNo=5000, PlotIt=FALSE, Verbose = FALSE, ...)

Arguments

DataOrDistances
Either nonsymmetric [1:n,1:d] datamatrix of n cases and d features or  
symmetric [1:n,1:n] distance matrix

ClusterNo
A number k which defines k different clusters to be built by the algorithm.

RandomNo
Only for "Steinley" or in case of distance matrix, number of random initial-  
izations with searching for minimal SSE, see [Steinley/Brusco, 2007]

Type
Choice of Kmeans algorithm, currently either "Hartigan" [Hartigan/Wong,  
1979], "LBG" [Linde et al., 1980], "Sparse" sparse k-means proposed in [Wit-  
ten/Tibshirani, 2010], "Steinley" best method of [Steinley/Brusco, 2007] pro-  
posed in Steinley 2003, "Lloyd" [Lloyd, 1982], "Forgy"[Forgy, 1965], MacQueen  
[MacQueen, 1967], or kcentroids [Leisch, 2006].

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, for kcentroids please see kcca  
of the flexclust package, or KMeansSparseCluster

Details

Uses either stats package function 'kmeans', cclust package implementation, flexclust package im- plementation or own code. In case of a distance matrix, RandomNo should be significantly lower  
than 5000, otherwise a long computation time is to be expected.

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 of the clustering algorithm used if existent, otherwise

SumDistsToCentroids: Vector of within-cluster sum of squares, one component per cluster

Centroids: the final cluster centers.

Note

The version using a distance matrix is still in the test phase and not yet verified.

Author(s)

Alfred Ultsch, Michael Thrun

References


Examples

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

data('Leukemia')
# As expected does not perform well
# For non-spherical cluster structures:
out=kmeansClustering(Leukemia$DistanceMatrix,ClusterNo=6,RandomNo =10,PlotIt=TRUE)
data('Hepta')
out=kmeansClustering(Hepta$Data,ClusterNo=7,
PlotIt=FALSE,Type="Steinley")

kmeansDist  k-means Clustering using a distance matrix

Description
Perform k-means clustering on a distance matrix

Usage
kmeansDist(Distance, ClusterNo=2,Centers=NULL,
RandomNo=1,maxIt = 2000,
PlotIt=FALSE,verbose = F)

Arguments
Distance  Distance matrix. For n data points of the dimension n x n
ClusterNo A number k which defines k different clusters to be built by the algorithm.
Centers  Default(NULL) a set of initial (distinct) cluster centres.
RandomNo  If>1: Number of random initializations with searching for minimal SSE is defined by this scalar
maxIt  Optional: Maximum number of iterations before the algorithm terminates.
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  Optional: Algorithm always outputs current iteration.

Value
Cls[1:n]  [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.
centerids[1:k] Indices of the centroids from which the cluster Cls was created

Note
Currently an experimental version

Author(s)
Felix Pape, Michael Thrun
Examples

data('Hepta')
#out=kmeansDist(as.matrix(dist(Hepta$Data)),ClusterNo=7,PlotIt=FALSE,RandomNo = 10)

## Not run:
data('Leukemia')
#as expected does not perform well
#for non-spherical cluster structures:
#out=kmeansDist(Leukemia$DistanceMatrix,ClusterNo=6,PlotIt=TRUE,RandomNo=10)

## End(Not run)

LargeApplicationClustering

Description

Clustering Large Applications (clara) of [Rousseeuw/Kaufman, 1990, pp. 126-163]

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 of d-dimensional data points. Every case has d attributes, variables or features.

ClusterNo A number k which defines k different clusters to be built 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


Examples

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

Description

Data is anonymized. Original dataset was published in [Haferlach et al., 2010]. Original dataset had around 12,000 dimensions. Detailed description 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


Examples

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

Lsun3D

Lsun3D inspired by FCPS introduced in [Thrun, 2018]

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 defines discontinuities, where the clusters have different variances. Three main clusters, and four outliers (in cluster 4). For a more detailed description see [Thrun, 2018].

References


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(DataOrDistances=NULL, Adjacency=NULL,
Radius=TRUE, DistanceMethod="euclidean", addLoops = TRUE, PlotIt=FALSE,...)

Arguments

DataOrDistances
NULL or: Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] not symmetric data matrix of n cases and d variables

Adjacency
Used if Data is NULL, matrix [1:n,1:n] defining which points are adjacent to each other by the number 1; not adjacent: 0

Radius
Scalar, 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] if Data instead of Distances are given.

DistanceMethod
Optional distance method of data, default is euclid, see parDist for details
\textit{MarkovClustering}

\begin{itemize}
\item \textbf{addLoops} Logical; if TRUE, self-loops with weight 1 are added to each vertex of x (see \texttt{mcl} of CRAN package \texttt{MCL}).
\item \textbf{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 \texttt{Cls}.
\item \textbf{...} Further arguments to be set for the clustering algorithm, if not set, default arguments are used.
\end{itemize}

\textbf{Details}

\texttt{DataOrDistances} is used to compute the Adjacency matrix if this input is missing. Then a unit-disk (R-ball) graph is calculated.

\textbf{Value}

\begin{itemize}
\item \textbf{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.
\item \textbf{Object} Object defined by clustering algorithm as the other output of this algorithm.
\end{itemize}

\textbf{Author(s)}

Michael Thrun

\textbf{References}


\textbf{Examples}

\begin{verbatim}
data('Hepta')
out=MarkovClustering(Data=Hepta$Data,PlotIt=FALSE)
\end{verbatim}
MinimalEnergyClustering

**Minimal Energy Clustering**

**Description**
Hierarchical 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 of 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', 'manhattan' (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.

- **...**
  
  In case of plotting further argument for `plot`, see `as.dendrogram`

**Value**

List of

- **Cls**
  
  If ClusterNo>0: [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. Otherwise ClusterNo=0: NULL

- **Dendrogram**
  
  Dendrogram of hierarchical clustering algorithm

- **Object**
  
  Ultrametric tree of hierarchical clustering algorithm

**Author(s)**

Michael Thrun

**References**

MinimaxLinkageClustering

See Also

HierarchicalClustering

Examples

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

MinimaxLinkageClustering

Minimax Linkage Hierarchical Clustering

Description

In the minimax 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), 'jaccard', '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.

... In case of plotting further argument for plot, see as.dendrogram

Value

List of

Cls

If, ClusterNo>0: [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. Otherwise for ClusterNo=0: NULL

Dendrogram

Dendrogram of hierarchical clustering algorithm

Object

Ultrametric tree of hierarchical clustering algorithm
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 of d-dimensional data points. Every case has d attributes, variables or features.
ClusterNo  A number k which defines k different clusters to be built 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

Model Based Clustering with Variable Selection

Description

Model-based clustering with variable selection and estimation of the number of clusters which is either based on [Marbac/Sedki, 2017],[Marbac et al., 2020], or on [Scrucca and Raftery, 2014].

Usage

ModelBasedVarSelClustering(Data,ClusterNo,Type,PlotIt=FALSE, ...)

Value

List of

Clss [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


See Also

MoGclustering

Examples

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

ModelBasedVarSelClustering

Model Based Clustering with Variable Selection

Usage

ModelBasedVarSelClustering(Data,ClusterNo,Type,PlotIt=FALSE, ...)

Note

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

Author(s)

Michael Thrun

References


See Also

MoGclustering

Examples

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

ModelBasedVarSelClustering

Model Based Clustering with Variable Selection

Description

Model-based clustering with variable selection and estimation of the number of clusters which is either based on [Marbac/Sedki, 2017],[Marbac et al., 2020], or on [Scrucca and Raftery, 2014].

Usage

ModelBasedVarSelClustering(Data,ClusterNo,Type,PlotIt=FALSE, ...)

Note

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

Author(s)

Michael Thrun

References


See Also

MoGclustering

Examples

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

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

ClusterNo Numeric which defines number of cluster to search for.

Type String, either VarSelLCM [Marbac/Sedki, 2017],[Marbac et al., 2020], or clustvarsel [Scrucca and Raftery, 2014].

PlotIt (optional) Boolean. Default = FALSE = No plotting performed.

... Further arguments passed on to VarSelCluster or clustvarsel.

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)

Quirin Stier, Michael Thrun

References


Examples

# Hepta
data("Hepta")
Data = Hepta$Data
V = ModelBasedVarSelClustering(Data, ClusterNo=7,Type="VarSelLCM")
Cls = V$Cls
ClusterAccuracy(Hepta$Cls, Cls, K = 7)

V = ModelBasedVarSelClustering(Data, ClusterNo=7,Type="clustvarsel")
Cls = V$Cls
ClusterAccuracy(Hepta$Cls, Cls, K = 7)

## Not run:
# Hearts
heart=VarSelLCM::heart
ztrue <- heart[,"Class"]
Data <- heart[,,-13]
Mixture of Gaussians Clustering using EM

Description
MixtureOfGaussians (MoG) clustering based on Expectation Maximization (EM) of [Chen et al., 2012] or algorithms closely resembling EM of [Benaglia/Chauveau/Hunter, 2009].

Usage
MoGclustering(Data,ClusterNo=2,Type,PlotIt=FALSE,Silent=TRUE,...)

Arguments
- Data: [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.
- ClusterNo: A number k which defines k different clusters to be built by the algorithm.
- Type: string defining approach to select: initialization approach of "EM" or "kmeans" of [Chen et al., 2012], or other methods "mvnormalmixEM" [McLachlan/Peel, 2000], "npEM"[Benaglia et al., 2009] or its extension "mvnpEM" [Chauveau/Hoang, 2016].
- 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: (optional) Boolean: print output or not (Default = FALSE = no output)
- ...: Further arguments to be set for the clustering algorithm, if not set, default arguments are used, see package mixtools EMCluster or mixtools for details.

Details
Algorithms for clustering through EM or its close resembles.

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. Type="mvnormalmixEM" sometimes fails

Author(s)

Michael Thrun

References


See Also

`ModelBasedClustering`

Examples

data('Hepta')
Data = Hepta$Data
out=MoGclustering(Data,ClusterNo=7,Type="EM",PlotIt=FALSE)
V=out$Cls

V1 = MoGclustering(Data,ClusterNo=7,Type="mvnpEM")
Cls1 = V1$Cls

V2 = MoGclustering(Data,ClusterNo=7,Type="npEM")
Cls2 = V2$Cls

## Not run:
#does not work always
V3 = MoGclustering(Data,ClusterNo=7,Type="mvnormalmixEM")
Cls3 = V3$Cls

## End(Not run)
MSTclustering

MST-kNN clustering algorithm [Inostroza-Ponta, 2008].

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

MSTclustering(DataOrDistances, DistanceMethod = "euclidean", PlotIt=FALSE, ...)

Arguments

DataOrDistances
Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] not symmetric data matrix of n cases and d variables

DistanceMethod
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

See Also

mst.knn

Examples

```r
data(Hepta)
MSTclustering(Hepta$Data)
```

Description

Either leiden [Traag et al., 2019] or louvain [Blondel et al., 2008] clustering

Usage

```r
NetworkClustering(DataOrDistances=NULL, Adjacency=NULL,
Type="louvain", Radius=FALSE, PlotIt=FALSE,...)
```

Arguments

- **DataOrDistances**: NULL or: [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
- **Adjacency**: Used if DataOrDistances is NULL, matrix [1:n,1:n] defining which points are adjacent to each other by the number 1; not adjacent: 0
- **Type**: Either "louvain" or "leiden"
- **Radius**: Scalar, 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]
- **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

DataOrDistances is used to compute the Adjecency matrix if this input is missing. Then a unit-disk (R-ball) graph is calculated. Radius=TRUE only works if data matrix is given.
NeuralGasClustering

Value

List of [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

Note

leiden requires igraph package and an installed python version. automatic installation may not work. manual call in console has to be in this case conda install -c conda-forge leidenalg

Author(s)

Michael Thrun

References


Examples

data('Hepta')
#out=NetworkClustering(Hepta$Data,PlotIt=FALSE)

NeuralGasClustering Neural gas algorithm for clustering

Description

Neural gas clustering published by [Martinetz et al., 1993] and implemented by [Boddenhofer 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 of d-dimensional data points. Every case has d attributes, variables or features.

- **ClusterNo**
  A number k which defines k different clusters to be built 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**


**Examples**

```r
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**

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

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of 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 NULL, automatic estimation is done using insights of [ultsch, 2005].

RadiusThreshold Threshold to identify clusters (RadiusThreshold <= MaxRadius), if NULL 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]. If NULL, its 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


See Also

optics
Examples

data('Hepta')
out=OPTICSclustering(Hepta$Data,MaxRadius=NULL,RadiusThreshold=NULL,minPts=NULL,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. 68-125].

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 of 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 built 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].
pdfClustering

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


Examples

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


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

pdfClustering Probability Density Distribution Clustering

Description

Clustering via non parametric density estimation

Usage

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

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of 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.
Penalized Regression-Based Clustering

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

- **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


Examples

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

Description

Clustering is performed through penalized regression with grouping pursuit

Usage

```r
PenalizedRegressionBasedClustering(Data, FirstLambda, SecondLambda, Tau, PlotIt = FALSE, ...)
```
PenalizedRegressionBasedClustering

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

FirstLambda Set 1 for quadratic penalty based algorithm, 0.4 for revised ADMM.

SecondLambda The magnitude of grouping penalty.

Tau Tuning parameter: tau, related to grouping penalty.

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 for PRclust, enables also usage of [Pan et al., 2013].

Details

Parameters are rather challenging to choose.

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

Data matrix is internally transposed in order to fit the definition of the algorithm.

Author(s)

Michael Thrun

References


Examples

data(Hepta)
Data=Hepta$Data
out=PenalizedRegressionBasedClustering(Data,0.4,1,2,PlotIt=FALSE)
table(out$Cls,Hepta$Cls)
Cluster Identification using Projection Pursuit as described in [Hofmeyr/Pavlidis, 2019].

Description

Summarizes recent projection pursuit methods for clustering based on [Hofmeyr/Pavlidis, 2015], [Hofmeyr, 2016] and [Pavlidis et al., 2016].

Usage

projectionPursuitClustering(Data, ClusterNo, Type = "MinimumDensity", PlotIt = FALSE, PlotSolution = FALSE, ...)

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

ClusterNo A number k which defines k different clusters to be built by the algorithm.

Type Either MinimumDensity [Pavlidis et al., 2016] MaximumClusterbility [Hofmeyr/Pavlidis, 2015], or NormalisedCut [Hofmeyr, 2016] or KernelPCA [Hofmeyr/Pavlidis, 2019].

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

PlotSolution Plots the partitioning solution as a tree as described in ...

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

Details

The details of the options for projection pursuit and partitioning of data are defined in [Hofmeyr/Pavlidis, 2019].

"KernelPCA" uses additionally the package kernlab and is implemented as given in the fifth example on page 21, section "extension" of [Hofmeyr/Pavlidis, 2019].

The first idea of using non-PCA projections for clustering was published by [Bock, 1987] as an definition. However, to the knowledge of the author it was not applied to any data. The first systematic comparison to Projection-Pursuit Methods projectionPursuitClustering and AutomaticProjectionBasedClustering can be found in [Thrun/Ultsch, 2018]. For PCA-based clustering methods please see TandemClustering
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


Examples

data('Hepta')
out=ProjectionPursuitClustering(Hepta$Data,ClusterNo=7,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 of d-dimensional data points. Every case has d attributes, variables or features.

Radius Maximum radius of clusters. If NULL, 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


Examples

data('Hepta')
out=QTclustering(Hepta$Data,Radius=NULL,PlotIt=FALSE)
RobustTrimmedClustering

Robust Trimmed Clustering

Description

Robust Trimmed Clustering invented by [Garcia-Escudero et al., 2008] and implemented by [Fritz et al., 2012].

Usage

RobustTrimmedClustering(Data, ClusterNo,
Alpha=0.05,PlotIt=FALSE,...)

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.
ClusterNo A number k which defines k different clusters to be built 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
Alpha No trimming is done equals to alpha =0, otherwise proportion of datapoints to be trimmed. tclust uses 0.05 as default.
...

Further arguments to be set for the clustering algorithm, e.g. nstart (number of random initializations),iter.max (maximum number of concentration steps),restr and restr.fact described in details. If not set, default arguments are used.

Details

"This iterative algorithm initializes k clusters randomly and performs "concentration steps" in order to improve the current cluster assignment. The number of maximum concentration steps to be performed is given by iter.max. For approximately obtaining the global optimum, the system is initialized nstart times and concentration steps are performed until convergence or iter.max is reached. When processing more complex data sets higher values of nstart and iter.max have to be specified (obviously implying extra computation time). ... The larger restr.fact is chosen, the looser is the restriction on the scatter matrices, allowing for more heterogeneity among the clusters. On the contrary, small values of restr.fact close to 1 imply very equally scattered clusters. This idea of constraining cluster scatters to avoid spurious solutions goes back to Hathaway (1985), who proposed it in mixture fitting problems" [Fritz et al., 2012]. The type of constraint restr can be set to "eigen", "deter" or "sigma.". Please see tclust for further 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**


**Examples**

```r
data('Hepta')
out=RobustTrimmedClustering(Hepta$Data,ClusterNo=7,Alpha=0,PlotIt=FALSE)
```

---

**SharedNearestNeighborClustering**

*SNN clustering*

**Description**

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

**Usage**

```r
SharedNearestNeighborClustering(Data,Knn,
Radius,minPts,PlotIt=FALSE,
UpperLimitRadius,...)
```
**Arguments**

- **Data**
  
  [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of 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 NULL, 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]. if NULL, its 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**


**See Also**

sNNclust

**Examples**

```r
data('Hepta')
out=SharedNearestNeighborClustering(
  Hepta$Data, Knn=7,Radius=NULL,minPts=NULL,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 approximately as an cluster.

**Usage**

```r
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 of 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 be ignored if ClusterNo is not NULL.

- **ClusterNo**  
  Optional. A number k which defines k different clusters to be built 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 is a stochastic clustering approach and research indicates that online should be preferred (c.f. [Thrun, 2018, p.37]).

**Value**

- **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
SOTAclustering

Author(s)

Michael Thrun

References


Examples

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

Description

Self-organizing Tree Algorithm (SOTA) introduced by [Herrero et al., 2001].

Usage

SOTAclustering(Data, ClusterNo,PlotIt=FALSE,UnrestGrowth,...)

Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

ClusterNo A number k which defines k different clusters to be built 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

UnrestGrowth TRUE: forces the ClusterNo option to uphold. FALSE: enables the algorithm to find its own number of clusters, in this cases ClusterNo should contain a high number because it is internally set as the number of iterations which is either reached or the max diversity criteria is satisfied priorly.

... 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.

sotaObject Object defined by clustering algorithm as the other output of this algorithm
Note

*Luis Winckelman integrated several function from clValid because it’s ORPHANED.

Author(s)

Luis Winckelmann*, Vasyl Pihur, Guy Brock, Susmita Datta, Somnath Datta

References


Examples

```r
#Does Work
data("Hepta")
out=SOTAclustering(Hepta$Data,ClusterNo=7)
table(Hepta$Cls,out$Cls)

#Does not work well
data("Lsun3D")
out=SOTAclustering(Lsun3D$Data,ClusterNo=100,PlotIt=FALSE,UnrestGrowth=FALSE)
```

SparseClustering

### Sparse Clustering

**Description**

Implements the sparse clustering methods of [Witten/Tibshirani, 2010].

**Usage**

```r
SparseClustering(DataOrDistances, ClusterNo, Type="Hierarchical",
PlotIt=F,Silent=FALSE, NoPerms=10,Wbounds, ...)
```

**Arguments**

- **DataOrDistances**
  
  Either a [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.
  
  or a [1:n,1:n] symmetric distance matrix.

- **ClusterNo**
  
  Numeric indicating number to cluster to find in Tree/ Dendrogramm in case of Type="Hierarchical" or numer of cluster to use in Type="kmeans"
Type (optional) Char selecting methods Hierarchical or kmeans. Default: "Hierarchical"
PlotIt (optional) Boolean. Default = FALSE = No plotting performed.
Silent (optional) Boolean: print output or not (Default = FALSE = no output)
NoPerms (optional), numeric scalar, Number of permutations.
Wbounds (optional) numeric vector, range of tuning parameters to consider. This is the L1 bound on w, the feature weights [Witten/Tibshirani, 2010].

Further arguments passed on to sparcl HierarchicalSparseCluster or KMeansSparseCluster depending on Type.

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
Tree Object Tree if Type="Hierarchical" is used.

Note

Quality of clustering results varies between sparse hierarchical if data is given in comparison to the case that distances are given.

Author(s)

Quirin Stier, Michael Thrun

References


Examples

# Hepta
data("Hepta")
Data = Hepta$Data
V1 = SparseClustering(Data, ClusterNo=7, Type="kmeans")
Cls1 = V1$Cls

V2 = SparseClustering(Data, ClusterNo=7, Type="Hierarchical")
Cls2 = V2$Cls

InputDistances = parallelDist::parDist(Data, method="euclidean")
DistanceMatrix = as.matrix(InputDistances)
V3 = SparseClustering(DistanceMatrix, ClusterNo=7, Type="Hierarchical")
SpectralClustering

### Description
Clusters the Data into "ClusterNo" different clusters using the Spectral 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 of `d`-dimensional data points. Every case has `d` attributes, variables or features.
- **ClusterNo** A number `k` which defines `k` different clusters to be built 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`: Kernel method, 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 function, `stringdot` String kernel
  - `kpar`: Kernel parameter: 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.

---

```
Cls3 = V3$Cls

## Not run:
set.seed(1)
Data = matrix(rnorm(100*50),ncol=50)
y = c(rep(1,50),rep(2,50))
Data[y==1,1:25] = Data[y==1,1:25]+2

V1 = SparseClustering(Data, ClusterNo=2, Type="kmeans")
Cls1 = V1$Cls

## End(Not run)
```
Value

List of C1s [1:1] 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


Examples

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

Description

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

Usage

Spectrum(Data, Type = 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 of d-dimensional data points. Every case has d attributes, variables or features.

Type Type=1: default eigengap method (Gaussian clusters)
Type=2: multimodality gap method (Gaussian/ non-Gaussian clusters)
Type=3: Allows to setClusterNo

ClusterNo Optional, A number k which defines k different clusters to be built 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 parameters 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 for 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


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.

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


SubspaceClustering

Algorithms for Subspace clustering

Description

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

Usage

SubspaceClustering(Data,ClusterNo,DimSubspace,

Type='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 built by the proclus or orclus algorithm.

**DimSubspace**  
Numerical number defining the dimensionality in which clusters should be search in in the orclus algorithm, for proclus it is an optional parameter

**Type**  
'Orclus', subspace clustering based on arbitrarily oriented projected cluster generation [Aggarwal and Yu, 2000]  
'ProClus' ProClus algorithm for subspace clustering [Aggarwal/Wolf, 1999]  
'Clique' ProClus algorithm finds subspaces of high-density clusters [Agrawal et al., 1999] and [Agrawal et al., 2005]  
'SubClu' SubClu algorithm is a density-connected approach for subspace 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 inCls

**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 a factor 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 intervals for each dimension) and "tau" (Density Threshold), see DBSCAN

Details

Subspace clustering algorithms have the goal to finde one or more subspaces with the assumation that sufficient dimensionality reduction is dimensionality reduction without loss of information. Hence subspace clustering aums at finding a linear subspace sucht that the subspace contains as much predictive information as the input space. The subspace is usually higher than two but lower than the input space. In contrast, projection-based clustering AutomaticProjectionBasedClustering projects the data (nonlinear) into two dimensions and tries only to preverse relevant neighborhoods.

Value

**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


Examples

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

TandemClustering

Tandem Clustering

Description

Summarizes clustering methods that combine k-means and pca

Usage

TandemClustering(Data,ClusterNo,Type="Reduced",PlotIt=FALSE,...)
Arguments

Data [1:n,1:d] matrix of dataset to be clustered. It consists of n cases of d-dimensional data points. Every case has d attributes, variables or features.

ClusterNo A number k which defines k different clusters to be built by the algorithm.

Type Reduced: Reduced k-means (RKM) [De Soete/Carroll, 1994].
Factorial: Factorial k-mean (FKM) [Vichi/Kiers, 2001]
KernelPCA: Kernel PCA with minimum normalised cut hyperplanes [Hofmeyr/Pavlidis, 2019]

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 C1s

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

Details

If the ClusterNo exceeds the number of dimensions, than the function is called recursively with ClusterNo=2. In each iteration the cluster with the highest number of overall points is clustered again, until the number of clusters is met.

"KernelPCA" uses additionally the package kernlab and is implemented as given in the fifth example on page 18, section "extension" of [Hofmeyr/Pavlidis, 2019]

The first idea of using non-PCA projections for clustering was published by [Bock, 1987] as an definition. However, to the knowledge of the author it was not applied to any data. The first systematic comparison to Projection-Pursuit Methods ProjectionPursuitClustering and AutomaticProjectionBasedClustering can be found in [Thrun/Ultsch, 2018].

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


Examples

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

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


Examples

data(Target)
str(Target)
Tetra

Tetra introduced in [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

Examples
data(Tetra)
str(Tetra)

TwoDiamonds

TwoDiamonds introduced in [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


Examples

data(TwoDiamonds)
str(TwoDiamonds)

WingNut introduced in [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


Examples

data(WingNut)
str(WingNut)
Index

* ADMM
  PenalizedRegressionBasedClustering, 88
* ADPclustering
  ADPclustering, 5
* Accuracy
  ClusteringAccuracy, 27
* Affinity Propagation
  APclustering, 8
* Agglomerative Nesting
  AgglomerativeNestingClustering, 6
* Agglomerative
  GenieClustering, 55
  HierarchicalClusterData, 60
  MinimaxLinkageClustering, 75
* Atom
  Atom, 10
* AutomaticProjectionBasedClustering
  AutomaticProjectionBasedClustering, 10
* Bouldin
  ClusterDaviesBouldinIndex, 21
* Chainlink
  Chainlink, 13
* Cluster Challenge
  ClusterChallenge, 18
* Cluster Count
  ClusterCount, 19
* Cluster Dendrogram
  ClusterDendrogram, 22
* Cluster Normalize
  ClusterNormalize, 34
* Cluster Sampling
  ClusterEqualWeighting, 26
* ClusterApply
  ClusterApply, 16
* ClusterCount
  ClusterCount, 19
* ClusterCreateClassification
  ClusterCreateClassification, 20
* ClusterDendrogram
  ClusterDendrogram, 22
* ClusterEqualWeighting
  ClusterEqualWeighting, 26
* ClusterNoEstimation
  ClusterNoEstimation, 31
* ClusterNormalize
  ClusterNormalize, 34
* ClusterPlotMDS
  ClusterPlotMDS, 35
* ClusterRenameDescendingSize
  ClusterRenameDescendingSize, 39
* ClusterRename
  ClusterRedefine, 37
  ClusterRename, 38
* Clusterability
  ClusterabilityMDplot, 14
* Clustering
  ClusterChallenge, 18
  ClusterDaviesBouldinIndex, 21
  ClusterDunnIndex, 25
  ClusteringAccuracy, 27
  DBSCAN, 43
  DBSclusteringAndVisualization, 45
  EstimateRadiusByDistance, 53
  GenieClustering, 55
  HierarchicalClusterData, 60
  HierarchicalClusterDists, 61
  HierarchicalClustering, 63
  HierarchicalDBSCAN, 64
  kmeansClustering, 66
  kmeansDist, 68
  LargeApplicationClustering, 69
  MinimalEnergyClustering, 74
  MinimaxLinkageClustering, 75
  OPTICSclustering, 84
  pdfClustering, 87
* Consecutive Clustering
INDEX

ClusterNormalize, 34
* Create Cluster Classification
  ClusterCreateClassification, 20
* DBSCAN
  DBSCAN, 43
  HierarchicalDBSCAN, 64
* DBS
  DBSclusteringAndVisualization, 45
* DC-ADMM
  PenalizedRegressionBasedClustering, 88
* Databionic swarm
  DBSclusteringAndVisualization, 45
* DatabionicSwarm
  DBSclusteringAndVisualization, 45
* Davies Bouldin Index
  ClusterDaviesBouldinIndex, 21
* DaviesBouldinIndex
  ClusterDaviesBouldinIndex, 21
* Davies
  ClusterDaviesBouldinIndex, 21
* Dendrogram
  ClusterDendrogram, 22
* Density Peak Clustering
  DensityPeakClustering, 48
* Density Peak
  DensityPeakClustering, 48
* DensityPeakClustering
  DensityPeakClustering, 48
* Descending Clustering
  ClusterRenameDescendingSize, 39
* Distances
  HierarchicalClusterDists, 61
* Divisive Analysis Clustering
  DivisiveAnalysisClustering, 50
* DivisiveAnalysisClustering
  DivisiveAnalysisClustering, 50
* Dunn Index
  ClusterDunnIndex, 25
* DunnIndex
  ClusterDunnIndex, 25
* EM clustering
  MoGclustering, 79
* EngyTime
  EngyTime, 51
* Equal Weighting
  ClusterEqualWeighting, 26
* Estimation of Number of Clusters
  ClusterNoEstimation, 31
* Expectation Maximization
  MoGclustering, 79
* FCPS
  Atom, 10
  Chainlink, 13
  ClusterChallenge, 18
  EngyTime, 51
  FCPS-package, 4
  GolfBall, 56
  Hepta, 59
  Leukemia, 70
  Lsun3D, 71
  Spectrum, 101
  Target, 107
  Tetra, 108
  TwoDiamonds, 108
  WingNut, 109
* Fundamental Clustering Problems Suite
  FCPS-package, 4
* Generate Fundamental Clustering Problem
  ClusterChallenge, 18
* GolfBall
  GolfBall, 56
* HCLclustering
  HCLclustering, 57
* HDDC
  HDDClustering, 58
* Hard Competitive learning clustering
  HCLclustering, 57
* Hepta
  Hepta, 59
* Hierarchical Clustering
  Hierarchicalclustering, 63
* Hierarchical DBSCAN
  HierarchicalDBSCAN, 64
* HierarchicalClustering
  HierarchicalClustering, 63
* Hierarchical
  GenieClustering, 55
  HierarchicalClusterData, 60
  HierarchicalClusterDists, 61
  HierarchicalClustering, 63
  HierarchicalDBSCAN, 64
  MinimalEnergyClustering, 74
  MinimaxLinkageClustering, 75
* Large Application Clustering
* LargeApplicationClustering, 69
  * LargeApplicationClustering
    LargeApplicationClustering, 69
  * Lsun3D
    Leukemia, 70
    Lsun3D, 71
  * MCC
    ClusterMCC, 30
  * MDS
    ClusterPlotMDS, 35
  * MDplot
    ClusterabilityMDplot, 14
  * MSTclustering
    MSTclustering, 81
  * Markov Clustering
    MarkovClustering, 72
  * Markov
    MarkovClustering, 72
  * Matthews Correlation Coefficient
    ClusterMCC, 30
  * Matthews Correlation
    ClusterMCC, 30
  * Matthews
    ClusterMCC, 30
  * Minimal Energy
    MinimalEnergyClustering, 74
  * MinimalEnergy
    MinimalEnergyClustering, 74
  * Minimax Linkage
    MinimaxLinkageClustering, 75
  * Minimax
    MinimaxLinkageClustering, 75
  * Mixture Of Gaussians
    ModelBasedClustering, 76
  * MixtureOfGaussians
    ModelBasedClustering, 76
    MoGclustering, 79
  * MoG
    ModelBasedClustering, 76
    MoGclustering, 79
  * Model based clustering
    ModelBasedClustering, 76
  * Model-based clustering
    ModelBasedVarSelClustering, 77
  * Multidimensional scaling
    ClusterPlotMDS, 35
  * Network Clustering
    NetworkClustering, 82
  * Neural Gas
    NeuralGasClustering, 83
  * NeuralGasClustering
    NeuralGasClustering, 83
  * PAM
    PAMclustering, 86
  * PDE
    StatPDEdensity, 103
  * PPCI
    ProjectionPursuitClustering, 90
  * Pareto Density Estimation
    StatPDEdensity, 103
  * Partitioning Around Medoids
    PAMclustering, 86
  * Penalized Regression Based Clustering
    PenalizedRegressionBasedClustering, 88
  * PenalizedRegressionBasedClustering
    PenalizedRegressionBasedClustering, 88
  * Projection Based Clustering
    AutomaticProjectionBasedClustering, 10
  * Projection Method
    ClusterPlotMDS, 35
  * ProjectionPursuitClustering
    ProjectionPursuitClustering, 90
  * Projection
    ClusterPlotMDS, 35
  * QTClustering
    QTclustering, 91
  * Radius
    EstimateRadiusByDistance, 53
  * Rename Descending Cluster Size
    ClusterRenameDescendingSize, 39
  * Rk statistic
    ClusterMCC, 30
  * Robust Trimmed Clustering
    RobustTrimmedClustering, 93
  * RobustTrimmedClustering
    RobustTrimmedClustering, 93
  * SMOTE
    ClusterUpsamplingMinority, 41
  * SOM
    SOMclustering, 96
  * SOTAclustering
    SOTAclustering, 97
  * Self-organizing Tree Algorithm
INDEX

SOTAclustering, 97
* Shannon information
  ClusterShannonInfo, 40
* Shannon
  ClusterShannonInfo, 40
* SharedNearest Neighbor Clustering
  SharedNearestNeighborClustering, 94
* Sparse Clustering
  SparseClustering, 98
* Spectral Clustering
  SpectralClustering, 100
  Spectrum, 101
* SpectralClustering
  SpectralClustering, 100
* Spectrum
  Spectrum, 101
* Subspace Clustering
  SubspaceClustering, 103
* SubspaceClustering
  SubspaceClustering, 103
* Tandem Clustering
  TandemClustering, 105
* TandemClustering
  TandemClustering, 105
* Target
  Target, 107
* Tetra
  Tetra, 108
* TwoDiamonds
  TwoDiamonds, 108
* Variable Selection
  ModelBasedVarSelClustering, 77
* WingNut
  WingNut, 109
* agnes
  AgglomerativeNestingClustering, 6
* apclustering
  APclustering, 8
* benchmarking
  FCPS-package, 4
* clara
  LargeApplicationClustering, 69
* cluster analysis
  AgglomerativeNestingClustering, 6
  DBSclusteringAndVisualization, 45
* clustering
  AgglomerativeNestingClustering, 6
  FCPS-package, 4
  PAMclustering, 86
* cluster
  FCPS-package, 4
* data entropy
  EntropyOfDataField, 52
* data field
  EntropyOfDataField, 52
* data set
  FCPS-package, 4
* databiontic
  DBSclusteringAndVisualization, 45
* datasets
  Atom, 10
  Chainlink, 13
  EngyTime, 51
  GolfBall, 56
  Hepta, 59
  Leukemia, 70
  Lsun3D, 71
  Target, 107
  Tetra, 108
  TwoDiamonds, 108
  WingNut, 109
* density estimation
  StatPDEdensity, 103
* diana
  DivisiveAnalysisClustering, 50
* distances
  ClusterDistances, 24
  ClusterInterDistances, 29
  kmeansDist, 68
* dunn
  ClusterDunnIndex, 25
* entropy
  EntropyOfDataField, 52
* factor
  ClusterCreateClassification, 20
* fanny
  FannyClustering, 54
* fast search and find of density peaks
  ADPclustering, 5
* fuzzy clustering
  FannyClustering, 54
* generalized Umatrix
  DBSclusteringAndVisualization, 45
* ggproto density estimation
  StatPDEdensity, 103
* information
  ClusterShannonInfo, 40
* inter cluster
  ClusterInterDistances, 29
* inter cluster
  ClusterInterDistances, 29
* intra cluster
  ClusterDistances, 24
* intra cluster
  ClusterDistances, 24
* k-batch clustering
  SOMclustering, 96
* k-batch
  SOMclustering, 96
* kmeans Clustering
  kmeansClustering, 66
  kmeansDist, 68
* kmeansClustering
  kmeansClustering, 66
  kmeansDist, 68
* leiden
  NetworkClustering, 82
* louvain
  NetworkClustering, 82
* model-based clustering
  HDDClustering, 58
* mst
  MSTclustering, 81
* optics
  OPTICSclustering, 84
* over sampling
  ClusterUpsamplingMinority, 41
* over-sampling
  ClusterUpsamplingMinority, 41
* pdfClustering
  pdfClustering, 87
* snn
  SharedNearestNeighborClustering, 94
* som clustering
  SOMclustering, 96
* subspace
  HDDClustering, 58
* swarm
  DBSclusteringAndVisualization, 45
* up sampling
  ClusterUpsamplingMinority, 41
* upsampling

ClusterUpsamplingMinority, 41
adpclust, 6
ADPclustering, 5, 5, 48, 49
AgglomerativeNestingClustering, 6
agnes, 7
APclustering, 8
as.dendrogram, 61, 62, 74, 75
Atom, 10
AutomaticProjectionBasedClustering, 10, 10, 90, 104, 106
Chainlink, 13
clara, 69
ClusterabilityMDplot, 14
ClusterAccuracy, 30, 31
ClusterAccuracy (ClusteringAccuracy), 27
ClusterApply, 16
ClusterChallenge, 18
ClusterCount, 19
ClusterCreateClassification, 20
ClusterDaviesBouldinIndex, 21
ClusterDendrogram, 22
ClusterDistances, 24
ClusterDunnIndex, 25
ClusterEqualWeighting, 26
ClusteringAccuracy, 27
ClusteringAlgorithms (FCPS-package), 4
ClusterInterDistances, 25, 29
ClusterIntraDistances
  (ClusterDistances), 24
ClusterMCC, 28, 30
ClusterNoEstimation, 31
ClusterNormalize, 34, 39
ClusterPlotMDS, 18, 19, 35, 42
ClusterRedefine, 37
ClusterRename, 38
ClusterRenameDescendingSize, 34, 39
ClusterShannonInfo, 40
ClusterUpsamplingDescendingSize, 41
clusvars, 78
cutree, 23

DatabionicSwarmClustering
  (DBSclusteringAndVisualization), 45
DBSCAN, 43, 53, 104
DBscan (DBSCAN), 43
DBSclustering, 46
INDEX

DBSclusteringAndVisualization, 45
DensityClust, 48, 49
DensityPeakClustering, 5, 6, 48
dist, 8
DivisiveAnalysisClustering, 50
EngyTime, 51
EntropyOfDataField, 52
EstimateRadiusByDistance, 53
FannyClustering, 54
FCPS-package, 4
GeneratePmatrix, 53
GeneratePswarmVisualization, 46
GenieClustering, 55, 63
GolfBall, 56
Hierarchical_DBSCAN (HierarchicalDBSCAN), 64
Hierarchical_DBscan (HierarchicalDBSCAN), 64
HierarchicalCluster
(HierarchicalClusterData), 60
HierarchicalClusterData, 58, 61, 62–64
HierarchicalClusterDists, 61, 61, 62–64
HierarchicalClustering, 33, 36, 60–62, 63, 75, 76
HierarchicalDBSCAN, 63, 64
HierarchicalSparseCluster, 99
index.DB, 21, 22
InterClusterDists
(ClusterInterDists), 29
IntraClusterDists
(ClusterDists), 24
kmeansClustering, 66
kmeansDist, 68
KMeansSparseCluster, 66, 99
LargeApplicationClustering, 69
Leukemia, 70
Lsun3D, 71
MarkovClustering, 72
MDplot, 16, 25, 30
MinimalEnergyClustering, 63, 64, 74
MinimaxLinkageClustering, 75
ModelBasedClustering, 76, 77, 80
ModelBasedVarSelClustering, 77
MoGclustering, 77, 79
mst.knn, 81, 82
MSTclustering, 81
NetworkClustering, 82
NeuralGasClustering, 83
optics, 85
OPTICSclustering, 84
PAMClustering (PAMclustering), 86
PAMclustering, 86
parDist, 48, 55, 61, 72, 74, 75, 81
pdfClustering, 87
PenalizedRegressionBasedClustering, 88
Plot3D, 35, 36
plot_ly, 49
plotTopographicMap, 12
PRclust, 89
ProjectionPursuitClustering, 12, 90, 90,
106
Pswarm, 46
QTClustering (QTclustering), 91
QTclustering, 91
RobustTrimmedClustering, 93
SharedNearestNeighborClustering, 94
similarities, 8, 9
sNNclust, 95
SOMclustering, 96
somgrid, 96
SOTAclustering, 97
sotaClustering (SOTAclustering), 97
SparseClustering, 63, 98
SpectralClustering, 100
Spectrum, 101, 102
StatPDEdensity, 103
SubspaceClustering, 12, 103
supersom, 96
TandemClustering, 12, 90, 105
Target, 107
tclust, 93
Tetra, 108
TwoDiamonds, 108
VarSelCluster, 78
WingNut, 109