Package ‘FCPS’

September 7, 2020

Type Package

Title Fundamental Clustering Problems Suite

Version 1.2.4

Date 2020-09-06

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

Description Many conventional clustering algorithms are provided in this package with consistent input and output, which enables the user to try out algorithms swiftly. Additionally, 26 statistical approaches for the estimation of the number of clusters as well as the the mirrored density plot (MD-plot) of clusterability are implemented. 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, mstknclusluster, densityClust, parallel, energy, R.utils, tclust, Spectrum, genie, proteoclust, fastcluster, clusterability, signal, reshape2, PFCI, clustrd, smacof, rgl.prclusluster, CEC, dendextend, moments,prabclus, knitr, rmarkdown

Depends R (>= 3.5.0)

License GPL-3

LazyData TRUE

LazyLoad yes

URL http://www.deepbionics.org

BugReports https://github.com/Mthrun/FCPS/issues

Encoding UTF-8

VignetteBuilder knitr
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: 2020-09-07 10:30:02 UTC

R topics documented:

FCPS-package ..................................................... 3
ADPclustering ...................................................... 5
AgglomerativeNestingClustering ................................. 6
APclustering ......................................................... 8
Atom ................................................................. 9
AutomaticProjectionBasedClustering ............................ 10
Chainlink .......................................................... 12
ClusterabilityMDplot ............................................... 13
ClusterApply ......................................................... 16
ClusterChallenge .................................................. 17
ClusterCount ......................................................... 19
ClusterCreateClassification ...................................... 20
ClusterDendrogram .................................................. 21
ClusterDistances .................................................... 22
ClusteringAccuracy .................................................. 23
ClusterInterDistances .............................................. 24
ClusterNoEstimation ............................................... 26
ClusterPlotMDS ....................................................... 28
ClusterRename ....................................................... 30
ClusterRenameDescendingSize ..................................... 31
CrossEntropyClustering ............................................ 32
DBscan .............................................................. 33
DBSclusteringAndVisualization .................................. 35
DensityPeakClustering .............................................. 37
DivisiveAnalysisClustering ....................................... 39
EngyTime ............................................................ 41
EntropyOfDataField ................................................ 41
EstimateRadiusByDistance ....................................... 42
FannyClustering .................................................... 43
GenieClustering ..................................................... 45
GolfBall ............................................................. 46
<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCLclustering</td>
<td>47</td>
</tr>
<tr>
<td>Hepta</td>
<td>48</td>
</tr>
<tr>
<td>HierarchicalClusterData</td>
<td>48</td>
</tr>
<tr>
<td>HierarchicalClusterDists</td>
<td>49</td>
</tr>
<tr>
<td>HierarchicalClustering</td>
<td>50</td>
</tr>
<tr>
<td>Hierarchical_DBSCAN</td>
<td>52</td>
</tr>
<tr>
<td>kmeansClustering</td>
<td>53</td>
</tr>
<tr>
<td>kmeansDist</td>
<td>55</td>
</tr>
<tr>
<td>LargeApplicationClustering</td>
<td>56</td>
</tr>
<tr>
<td>Leukemia</td>
<td>58</td>
</tr>
<tr>
<td>Lsun3D</td>
<td>59</td>
</tr>
<tr>
<td>MarkovClustering</td>
<td>59</td>
</tr>
<tr>
<td>MinimalEnergyClustering</td>
<td>61</td>
</tr>
<tr>
<td>MinimaxLinkageClustering</td>
<td>62</td>
</tr>
<tr>
<td>ModelBasedClustering</td>
<td>63</td>
</tr>
<tr>
<td>MoGclustering</td>
<td>65</td>
</tr>
<tr>
<td>MSTclustering</td>
<td>66</td>
</tr>
<tr>
<td>NeuralGasClustering</td>
<td>67</td>
</tr>
<tr>
<td>OPTICSclustering</td>
<td>68</td>
</tr>
<tr>
<td>PAMclustering</td>
<td>70</td>
</tr>
<tr>
<td>pdfClustering</td>
<td>71</td>
</tr>
<tr>
<td>PenalizedRegressionBasedClustering</td>
<td>72</td>
</tr>
<tr>
<td>ProjectionPursuitClustering</td>
<td>74</td>
</tr>
<tr>
<td>QTclustering</td>
<td>75</td>
</tr>
<tr>
<td>RobustTrimmedClustering</td>
<td>77</td>
</tr>
<tr>
<td>SharedNearestNeighborClustering</td>
<td>78</td>
</tr>
<tr>
<td>SOMclustering</td>
<td>79</td>
</tr>
<tr>
<td>SOTAclustering</td>
<td>81</td>
</tr>
<tr>
<td>SpectralClustering</td>
<td>82</td>
</tr>
<tr>
<td>Spectrum</td>
<td>83</td>
</tr>
<tr>
<td>StatPDEdensity</td>
<td>85</td>
</tr>
<tr>
<td>SubspaceClustering</td>
<td>85</td>
</tr>
<tr>
<td>TandemClustering</td>
<td>87</td>
</tr>
<tr>
<td>Target</td>
<td>89</td>
</tr>
<tr>
<td>Tetra</td>
<td>90</td>
</tr>
<tr>
<td>TwoDiamonds</td>
<td>90</td>
</tr>
<tr>
<td>WingNut</td>
<td>91</td>
</tr>
</tbody>
</table>

**Index**

`92`
Description

Many conventional 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. 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 'Fundamental Clustering Problems Suite' (FCPS) originally 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 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 and provides a standardized and easy access to many clustering algorithms.

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.

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

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

Details

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

Value

List of

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

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

Author(s)

Michael Thrun
Agglomerative Nesting Clustering

References


See Also

DensityPeakClustering

adpclust

Examples

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

Agglomerative Nesting Clustering

AGNES clustering

Description

Agglomerative hierarchical clustering (AGNES)

Usage

AgglomerativeNestingClustering(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 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
AgglomerativeNestingClustering

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

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

ClusterDendrogram(CA$Dendrogram,7,main='AGNES clustering')

print(CA$Object)
plot(CA$Object)

---

**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.

- **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/

See Also

apcluster

Examples

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

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

---

Atom Atom introduced in [Ultsch, 2004].

Description

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

Usage

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

Details

Size 800, Dimensions 3, stored in Atom$Data

Classes 2, stored in Atom$Cls

References


Examples

data(Atom)
str(Atom)

AutomaticProjectionBasedClustering

Automatic Projection-Based Clustering

Description

Performs projection-based clustering without user-interaction based on projection methods. The approach is published in [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.
AutomaticProjectionBasedClustering

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StructureType</td>
<td>Either compact (TRUE) or connected (FALSE), see discussion in [Thrun, 2018]</td>
</tr>
<tr>
<td>PlotIt</td>
<td>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</td>
</tr>
<tr>
<td>PlotTree</td>
<td>Plots the dendrogram</td>
</tr>
<tr>
<td>PlotMap</td>
<td>Plots the topographic map [Thrun et al., 2016].</td>
</tr>
<tr>
<td>...</td>
<td>Further arguments to be set for the clustering algorithm, if not set, default arguments are used.</td>
</tr>
</tbody>
</table>

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, which exploited the generalized U-Matrix and was introduced in [Thrun/Ultsch, 2017]. 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 clustering, improve it or set the number of clusters appropriately. However, 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 and CA-based clustering methods can be found in [Thrun/Ultsch, 2020a]. 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=AutomaticProjectionBasedClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

---

Chainlink

Chainlink introduced in [Ultsch et al., 1994; Ultsch, 1995].

Description

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

Usage

data("Chainlink")

Details

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

References


Examples

data(Chainlink)
str(Chainlink)

### 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], pvalues above 0.05 can be given for MDplots which are clearly multimodal.

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

[Adolfsson et al., 2019] Adolfsson, A., Ackerman, M., & Brownstein, N. C.: To cluster, or not to

[Thrun et al., 2020] Thrun, M. C., Gehlert, T. & Ultsch, A.: Analyzing the Fine Structure of Dis-


Mirrored-Density Plot, in Archambault, D., Nabney, I. & Peltonen, J. (eds.), Machine Learning

See Also

MDplot

Examples

#one dataset
data(Hepta)
ClusterabilityMDplot(Hepta$Data)

#multiple datasets
data(Atom)
data(Chainlink)
data(Lsun3D)
data(GolfBall)
data(EngyTime)
data(Target)
data(Tetra)
data(WingNut)
data(TwoDiamonds)
DataV = list(
   Atom = Atom$Data,
   Chainlink = Chainlink$Data,
   Hepta = Hepta$Data,
   Lsun3D = Lsun3D$Data,
   GolfBall = GolfBall$Data,
   EngyTime = EngyTime$Data,
   Target = Target$Data,
   Tetra = Tetra$Data,
   WingNut = WingNut$Data,
   TwoDiamonds = TwoDiamonds$Data
)
ClusterabilityMDplot(DataV)
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,...)

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.
- ...: Additional parameters to be passed on to FUN.

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 distances 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

```r
# 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)
```

```r
## Not run:
Iris=datasets::iris
Distances=as.matrix(Iris[,1:4])
SomeFactors=Iris$Species
V=ClusterCreateClassification(SomeFactors)
Cls=V$Cls
V$ClusterNames
ClusterApply(Distances,mean,Cls)
```

```r
## Not run:
Iris=datasets::iris
Distances=as.matrix(Iris[,1:4])
SomeFactors=Iris$Species
V=ClusterCreateClassification(SomeFactors)
Cls=V$Cls
V$ClusterNames
ClusterApply(Distances,mean,Cls)
```

```r
## Not run:
suppressPackageStartupMessages(library("prabclus", quietly = TRUE))
data(tetragonula)
# Generated Specific Distance Matrix
ta <- alleleconvert(strmatrix=as.matrix(tetragonula[1:236,]))
tai <- alleleinit(allelematrix=ta,distance="none")
Distance=alleledist((unbuild.charmatrix(tai$charmatrix,236,13)),236,13)
MDStrans=ClusterApply(Distance,identity)$identityPerCluster
```

```r
## End(Not run)
```

---

**Description**

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

**Usage**

```r
ClusterChallenge(Name,SampleSize,
PlotIt=FALSE,PointSize=1,Plotter3D="rgl",...)
```
### Arguments

- **Name**: string, either 'Atom', 'Chainlink', 'EngyTime', 'GolfBall', 'Hepta', 'Lsun3D', 'Target' 'Tetra' 'TwoDiamonds' 'WingNut
- **SampleSize**: Size of Sample higher than 300, preferable above 500
- **PlotIt**: TRUE: Plots the challenge with `ClusterPlotMDS`
- **PointSize**: If PlotIt=TRUE: see `ClusterPlotMDS`
- **Plotter3D**: If PlotIt=TRUE: see `ClusterPlotMDS`
- ... If PlotIt=TRUE: further arguments for `ClusterPlotMDS`

### 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

```r
ClusterChallenge("Chainlink",2000,TRUE)
```
**ClusterCount**

**Description**

Calculates statistics for clustering in each group of the data points.

**Usage**

ClusterCount(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**

The ordering of the output is defined by the first occurrence of every cluster in Cls. If non finite values are given in the numerical vector, they are set to the "9999" cluster.

**Value**

- **UniqueClasses**: The unique clusters in Cls.
- **CountPerClass**: The number of data points in the corresponding unique clusters.
- **NumberOfClasses**: 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)
ClusterCreateClassification

Create Classification for Cluster. functions

Description
Creates a Cls from arbitrary list of objects

Usage
ClusterCreateClassification(Objects)

Arguments
Objects
Listed objects, for example factor

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 belongs 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)
```
Description

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

Usage

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

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

ClusterDistances

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

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

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

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

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

Author(s)
Michael Thrun
ClusteringAccuracy

References


See Also

MDplot
ClusterInterDistances

Examples

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

interdists=ClusterDistances(Distance,Hepta$Cls)

ClusteringAccuracy ClusterAccuracy

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].

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**

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


**Examples**

```r
# 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)

data('Hepta')
Cls=kmeansClustering(Hepta$Data,7,Type = "Hartigan",nstart=100)
table(Cls$Cls,Hepta$Cls)
ClusterAccuracy(Hepta$Cls,Cls$Cls)
```

**Description**

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

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

Arguments

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

Cls [1:n] numerical vector of 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.

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 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)
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 classes is given for each indicator.

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

Usage

ClusterNoEstimation(DataOrDistances, ClsMatrix = NULL, max.nc, index = "all", min.nc = 2, Silent = TRUE, method = NULL, PlotIt=TRUE,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:(max.nc)] Clustering of the number of classes to be checked as a matrix with one cluster per column (see also details (2) and (3)), must be specified if method = NULL

max.nc
Highest number of classes to be checked

method
Cluster procedure, with which the clusterings are created (see details (4) for possible methods), must be specified if ClsMatrix = NULL

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

min.nc
Lowest number of classes to be checked, default = 2

Silent
If TRUE status messages are output, default = FALSE

PlotIt
If TRUE plots fanplot with proposed cluster numbers

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 max.nc+1 classes. If these key figures are to be calculated, this clustering must be specified in cls.
(3)
The indicator kl requires a clustering for min.nc-1 classes. If this key figure is to be calculated, this clustering must also be specified in cls. For the case min.nc = 2 no clustering for 1 has to be given.
(4)
The following methods can be used to create clusterings:
(5)
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,min.nc:(max.nc)] 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

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

Author(s)

Peter Nahrgang
References


Examples

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

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

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

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

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(DataOrDists, Cls, main = "Clustering",
method = "euclidean", OutputDimension = 3,
PointSize=1,Plotter3D="rgl",Colorsequence, ...)

Arguments

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

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

method 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 dimensions, 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
ClusterRename

Examples

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

```r
data(Leukemia)
ClusterPlotMDS(Leukemia$DistanceMatrix, Leukemia$Cls)
```

---

### Description

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

### Usage

```r
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
ClusterRenameDescendingSize

Examples

```r
data('Hepta')
Cls=Hepta$Cls
Data=Hepta$Data#

#prior
Cls

#Named Clustering
ClusterRename(Cls, Data)
```

Description

Renames the clusters of a classification in descending order.

Usage

```r
ClusterRenameDescendingSize(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.

Value

- **RenamedCls**
  - The renamed classification. A vector of clusters, where the largest cluster is `C1` and so forth.

Author(s)

Michael Thrun, Alfred Ultsch

Examples

```r
data('Lsun3D')
Cls=Lsun3D$Cls

# not descending cluster numbers
Cls[Cls==1]=543
Cls[Cls==4]=1

# Now ordered by cluster size and descending
ClusterRenameDescendingSize(Cls)
```
CrossEntropyClustering

Cross-Entropy Clustering

Description

Neural gas clustering published by [Tabor/Spurek, 2014] and implemented by [Spurek et al., 2017].

Usage

CrossEntropyClustering(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.

Details

Contrary to most of the other implemented algorithms in this package, the results on the easiest clustering challenge of Hepta are unstable for cross-entropy clustering in the sense that the clustering is not always correct. Reproducibility experiments should be performed (see [Tabor/Spurek, 2014]).

Value

List of

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

Author(s)

Michael Thrun

References

DBscan

Examples

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

DBscan

Description
DBscan clustering

Usage

DBscan(Data,Radius,minPts,

PlotIt=FALSE,UpperLimitRadius,...)

Arguments

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

#search for right parameter setting by grid search
data("WingNut")
Data = WingNut$Data
DBSGrid <- expand.grid(
  Radius = seq(from = 0.01, to = 0.3, by = 0.02),
  minPTs = seq(from = 1, to = 50, by = 2)
)
BestAcc = c()
for (i in seq_len(nrow(DBSGrid))) {
  print(i)
  parameters <- DBSGrid[i,]
  Cls9 = DBscan(
    Data,
    minPts = parameters$minPTs,
    Radius = parameters$Radius,
    PlotIt = F,
    UpperLimitRadius = parameters$Radius
  )$Cls
  if (length(unique(Cls9)) < 5)
    BestAcc[i] = DatabionicSwarm::ClusteringAccuracy(WingNut$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
Description

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

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 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.

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

Cl1s 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.

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 = T),
    sample(1:100, 300, replace = T),
    sample(1:100, 300, replace = T)
)
# Make sure there are no structures
# (sample size is small and still could generate structures randomly)
```
DensityPeakClustering

Density Peak Clustering algorithm using the Decision Graph

Description

DensityPeakClustering

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

Arguments
DataOrDistances
Either [1:n,1:n] symmetric distance matrix or [1:n,1:d] 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
method
Optional distance method of data, default is euclid, see parDist for details
PlotIt
Optional TRUE: Plots 2d or 3d result with clustering
Data
[1:n,1:d] data matrix in the case that DataOrDistances is missing and partial
matching does not work.
...
Optional, further arguments for densityClust

Details
The densityClust algorithm does not decide the k number of clusters, this has to be done by the
parameter setting. This 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).

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
**DivisiveAnalysisClustering**

**References**


**See Also**

ADPclustering  
densityClust

**Examples**

```r
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
```

**Description**

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

**Usage**

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

DataOrDistances

[1:n,1:d] matrix of dataset to be clustered. It consists of n cases 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, the dendrogram is generated instead of a clustering to estimate the numbers of clusters.

PlotIt

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

Standardization

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

Data

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

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

**Value**

List of

Cls

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

Dendrogram

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

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

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

EntropyOfDataField(Data,

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

, PlotIt = TRUE)
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

Examples

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

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

Description

...

Usage

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

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

Author(s)

Michael Thrun

References


Examples

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**

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

**Arguments**

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

- ... furter argument to genie like:

- **thresholdGini**
  
  Single numeric value in [0,1], threshold for the Gini index, 1 gives the standard single linkage algorithm

**Details**

Wrapper for Genie algorithm.

**Value**

- **List of**

- **Clss**
  
  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

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

---

GolfBall

_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

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

Details

Size 4002, Dimensions 3, stored in GolfBall$Data

Classes 1, stored in GolfBall$Cls

References


Examples

```r
data(GolfBall)
str(GolfBall)
```
**HCLclustering**  
*On-line Update (Hard Competitive learning) method*

### Description

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

### Usage

```r
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

- **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=HCLclustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)
```
HierarchicalClusterData

Description

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

Usage

data("Hepta")

Details

Size 212, Dimensions 3, stored in Hepta$Data
Classes 7, stored in Hepta$Cls

References


Examples

data(Hepta)
str(Hepta)

HierarchicalClusterData

Hierarchical Clustering of Data

Description

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

Usage

HierarchicalClusterData(Data,ClusterNo=0,
method="ward.D2",DistanceMethod="euclidean",
ColorThreshold=0, Fast=FALSE,Cls=NULL,...)
HierarchicalClusterDists

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.

method Methode der Clusterung: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".

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

ColorTreshold Draws cutline w.r.t. 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

Examples

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

HierarchicalClusterDists

Hierarchical Clustering with Distances

Description

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

Usage
HierarchicalClusterDists(pDist, ClusterNo=0, method="ward.D2", ColorTreshold=0, Fast=FALSE, ...)

Arguments
- **pDist**: Distances as either matrix [1:n,1:n] or dist object
- **ClusterNo**: A number k which defines k different clusters to be built by the algorithm.
- **method**: Method of cluster analysis: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid".
- **ColorTreshold**: Draws cutline w.r.t. dendogram y-axis (height), height of line as scalar should be given
- **Fast**: If TRUE and fastcluster installed, then a faster implementation of the methods above can be used
- **...**: 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

Examples
```r
data("Hepta")
#out=HierarchicalClusterDists(as.matrix(dist(Hepta$Data)), ClusterNo=7)
```

Description
Wrapper for various agglomerative hierarchical clustering algorithms.
HierarchicalClustering

Usage

HierarchicalClustering(DataOrDistances, ClusterNo, method='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.

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

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") or Hierarchical_DBSCAN (for HDBSCAN).

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

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
Hierarchical_DBSCAN

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

Usage
Hierarchical_DBSCAN(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 argu-

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].
kmeansClustering

**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**

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

data('Leukemia')
set.seed(1234)
CA=Hierarchical_DBSCAN(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**

```r
kmeansClustering(DataOrDistance, ClusterNo,  
                  Type = 'LBG',RandomNo=5000, PlotIt=FALSE, Verbose = FALSE,... )
```
**Arguments**

- **DataOrDistance**: 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 initializations with searching for minimal SSE, see [Steinley/Brusco, 2007].
- **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,...

**Details**

Uses either stats package function 'kmeans', cclust package implementation 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 List V of**
  - **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

```r
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**

```r
kmeansDist(Distance, ClusterNo=2,Centers=NULL,
RandomNo=1,maxIt = 2000,
PlotIt=FALSE,verbose = F)
```
Large Application Clustering

**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**

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

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

---

LargeApplicationClustering

**Description**

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

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 C1s
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 argu-
ments are used.

Details

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

Value

List of

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

Author(s)

Michael Thrun

References

[Rousseeuw/Kaufman, 1990] Rousseeuw, P. J., & Kaufman, L.: Finding groups in data, Belgium,
Examples

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

Leukemia

<table>
<thead>
<tr>
<th>Leukemia distance matrix and classification used in [Thrun, 2018]</th>
</tr>
</thead>
</table>

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**

```r
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**

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

---

**MarkovClustering**

*Markov Clustering*

**Description**

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

**Usage**

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

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. This is used if Adjacency is missing. Then a unit-disk (R-ball) graph is calculated.

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

Radius Radius for unit disk graph (r-ball graph) if adjacency matrix is missing. Automatic estimation can be done either with =TRUE [Ultsch, 2005] or FALSE [Thrun et al., 2016]

addLoops Logical; if TRUE, self-loops with weight 1 are added to each vertex of x (see mcl of CRAN package MCL).

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

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

Details ...

Value ...

List of

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

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

Author(s)

Michael Thrun

References


Examples

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

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 rep-
senting 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


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),'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

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

Value

List of

Cl\(\text{s}\) If, \(\text{ClusterNo}>0\): \([1:n]\) 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 \(\text{ClusterNo}=0\): 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')
out=MinimaxLinkageClustering(Hepta$Data,ClusterNo=7)

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**


**Value**

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

**Note**

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

**Author(s)**

Michael Thrun

**References**


**See Also**

`MoGclustering`

**Examples**

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

Description

call MixtureOfGaussians (MoG) clustering based on Expectation Maximization (EM) of [Chen et al., 2012].

Usage

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

Arguments

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

method Initialization by either "EM" oder "kmeans"

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

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

Details

...

Value

List of

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

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

Note

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

Author(s)

Michael Thrun
References


See Also

ModelBasedClustering

Examples

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

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

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

Arguments

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

Details

Does not work on Hepta with euclidean distances.
Value

List of

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

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

Author(s)

Michael Thrun

References


See Also

mst.knn

Examples

data(Hepta)

MSTclustering(Hepta$Data)
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.
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

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=NeuralGasClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

OPTICScclustering

Role

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

Usage

OPTICScclustering(Data, MaxRadius,RadiusThreshold, minPts = 5, 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.

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=OPTICScustering(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. 164-198].

Usage

PAMclustering(DataOrDistances,ClusterNo,
               PlotIt=FALSE,Standardization=TRUE,Data,...)

Arguments

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

Details

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

List of

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

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

Author(s)

Michael Thrun

References


Examples

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.
Details

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

Value

List of

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

Author(s)

Michael Thrun

References


Examples

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

### Penalized Regression-Based Clustering

**Penalized Regression-Based Clustering of [Wu et al., 2016].**

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$Data
out=PenalizedRegressionBasedClustering(Data,0.4,1,2,FALSE)
table(out$Cls,Hepta$Cl1s)
ProjectionPursuitClustering

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

Cla [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, 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.

... Further arguments to be set for the clustering algorithm, e.g. restr and restr.fact described in details. If not set, default arguments are used.

Details

"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

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

SharedNearestNeighborClustering(Data,Knn=7,Radius,minPts,PlotIt=FALSE,
UpperLimitRadius,...)

Arguments

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

data('Hepta')
out=SharedNearestNeighborClustering(Hepta$Data,Radius=NULL,minPts=NULL,PlotIt = FALSE)

SOMclustering self-organizing maps based clustering implemented by [Werens, 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

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

List of

Cls [1:n] numerical vector defining the classification as the main output of the clustering algorithm

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

Author(s)

Michael Thrun

References


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 intergrated several function from clValid because it's ORPHANED.

Author(s)

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


Examples

#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)

SpectralClustering  Spectral Clustering

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 inCls

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

kernel : Kernel method, possible options: rbf dot Radial Basis kernel function "Gaussian" poly dot Polynomial kernel function vanilladot Linear kernel function tanhdot Hyperbolic tangent kernel function laplacedot Laplacian kernel function besseldot Bessel kernel function anovadot ANOVA RBF kernel function splinedot Spline kernel stringdot String kernel

kpar : 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.

**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=SpectralClustering(Hepta$Data,ClusterNo=7,PlotIt=FALSE)

---

**Spectrum**

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

**Description**

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

**Usage**

Spectrum(Data, Method = 2, ClusterNo = NULL,
PlotIt = FALSE, Silent = TRUE,PlotResults = FALSE, ...)

**Arguments**

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

*Method*  
Method=1: default eigengap method (Gaussian clusters)  
Method=2: multimodality gap method (Gaussian/non-Gaussian clusters)  
Method=3: Allows to setClusterNo
ClusterNo  Optional, A number k which defines k different clusters to be 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


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

Usage

SubspaceClustering(Data,ClusterNo,DimSubspace, method='Orclus',PlotIt=FALSE,OrclusInitialClustersNo=ClusterNo+2,...)
Arguments

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

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

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

method 'Orclus', subspace clustering based on arbitrarily oriented projected cluster generation [Aggarwal and Yu, 2000]
'ProClus' ProClus Algorithm for Projected Clustering [Aggarwal/Wolf, 1999]
'Clique' ProClus Algorithm for Projected Clustering [Agrawal et al., 1999] and [Agrawal et al., 2005]
'SubClu' ProClus Algorithm for Projected Clustering [Kailing et al., 2004]

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

OrclusInitialClustersNo Only for Orclus algorithm: Initial number of clusters (that are computed in the entire data space) must be greater than k. The number of clusters is iteratively decreased by 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

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

Value

List of

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

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

Note

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

Michael Thrun

**References**


Further "advertising" can be found in https://towardsdatascience.com/subspace-clustering-7b884e8fff73

**Examples**

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

---

**TandemClustering**

*Tandem Clustering*

**Description**

Summarizes clustering methods that combine k-means and pca

**Usage**

```r
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.
TandemClustering

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 Cls

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

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

WingNut

WingNut introduced in [Ultsch, 2005]

Description

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

Usage

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

Details

Size 1016, Dimensions 2, stored in WingNut$Data
Classes 2, stored in WingNut$Cls

References


Examples

```r
data(WingNut)
str(WingNut)
```
Index

* ADMM
  PenalizedRegressionBasedClustering, 72
* ADPclustering
  ADPclustering, 5
* Accuracy
  ClusteringAccuracy, 23
* Affinity Propagation
  APclustering, 8
* Agglomerative Nesting
  AgglomerativeNestingClustering, 6
* Agglomerative
  GenieClustering, 45
  HierarchicalClusterData, 48
  MinimaxLinkageClustering, 62
* Atom
  Atom, 9
* AutomaticProjectionBasedClustering
  AutomaticProjectionBasedClustering, 10
* Chainlink
  Chainlink, 12
* Cluster Challenge
  ClusterChallenge, 17
* Cluster Count
  ClusterCount, 19
* Cluster Dendrogram
  ClusterDendrogram, 21
* ClusterApply
  ClusterApply, 16
* ClusterCount
  ClusterCount, 19
* ClusterCreateClassification
  ClusterCreateClassification, 20
* ClusterDendrogram
  ClusterDendrogram, 21
* ClusterNoEstimation
  ClusterNoEstimation, 26
* ClusterPlotMDS
  ClusterPlotMDS, 28
* ClusterRenameDescendingSize
  ClusterRenameDescendingSize, 31
* ClusterRename
  ClusterRename, 30
* Clusterability
  ClusterabilityMDplot, 13
* Clustering
  ClusterChallenge, 17
  ClusteringAccuracy, 23
  DBscan, 33
  DBSclusteringAndVisualization, 35
  EstimateRadiusByDistance, 42
  GenieClustering, 45
  Hierarchical_DBSCAN, 52
  HierarchicalClusterData, 48
  HierarchicalClusterDists, 49
  HierarchicalClustering, 50
  kmeansClustering, 53
  kmeansDist, 55
  LargeApplicationClustering, 56
  MinimalEnergyClustering, 61
  MinimaxLinkageClustering, 62
  OPTICSclustering, 68
  pdfClustering, 71
* Create Cluster Classification
  ClusterCreateClassification, 20
* Cross-Entropy Clustering
  CrossEntropyClustering, 32
* Cross-Entropy
  CrossEntropyClustering, 32
* CrossEntropyClustering
  CrossEntropyClustering, 32
* DBSCAN
  Hierarchical_DBSCAN, 52
* DBS
  DBSclusteringAndVisualization, 35
* DBscan
  DBscan, 33
**DC-ADMM**  
PenalizedRegressionBasedClustering, 72

**DatabionicSwarm**  
DBSclusteringAndVisualization, 35

**Dendrogram**  
ClusterDendrogram, 21

**Density Peak Clustering**  
DensityPeakClustering, 37

**Density Peak**  
DensityPeakClustering, 37

**DensityPeakClustering**  
DensityPeakClustering, 37

**Descending Clustering**  
ClusterRenameDescendingSize, 31

**Distances**  
HierarchicalClusterDists, 49

**Divisive Analysis Clustering**  
DivisiveAnalysisClustering, 39

**DivisiveAnalysisClustering**  
DivisiveAnalysisClustering, 39

**EM clustering**  
MoGclustering, 65

**EngyTime**  
EngyTime, 41

**Estimation of Number of Clusters**  
ClusterNoEstimation, 26

**Expectation Maximization**  
MoGclustering, 65

**FCPS**  
Atom, 9  
Chainlink, 12  
ClusterChallenge, 17  
EngyTime, 41  
FCPS-package, 3  
GolfBall, 46  
Hepta, 48  
Leukemia, 58  
Lsun3D, 59  
Spectrum, 83  
Target, 89  
Tetra, 90  
TwoDiamonds, 90  
WingNut, 91

**Fundamental Clustering Problems Suite**  
FCPS-package, 3

**Generate Fundamental Clustering Problem**  
ClusterChallenge, 17

**GolfBall**  
GolfBall, 46

**HCLclustering**  
HCLclustering, 47

**Hard Competitive learning clustering**  
HCLclustering, 47

**Hepta**  
Hepta, 48

**Hierarchical Clustering**  
HierarchicalClustering, 50

**Hierarchical DBSCAN**  
Hierarchical_DBSCAN, 52

**HierarchicalClustering**  
HierarchicalClustering, 50

**Hierarchical**  
GenieClustering, 45  
Hierarchical_DBSCAN, 52  
HierarchicalClusteringData, 48  
HierarchicalClusterDists, 49  
HierarchicalClustering, 50  
MinimalEnergyClustering, 61  
MinimaxLinkageClustering, 62

**Large Application Clustering**  
LargeApplicationClustering, 56

**LargeApplicationClustering**  
LargeApplicationClustering, 56

**Lsun3D**  
Leukemia, 58  
Lsun3D, 59

**MDS**  
ClusterPlotMDS, 28

**MDplot**  
ClusterabilityMDplot, 13

**MSTclustering**  
MSTclustering, 66

**Markov Clustering**  
MarkovClustering, 59

**Markov**  
MarkovClustering, 59

**Minimal Energy**  
MinimalEnergyClustering, 61

**MinimalEnergy**  
MinimalEnergyClustering, 61

**Minimax Linkage**  
MinimaxLinkageClustering, 62

**Minimax**  
MinimaxLinkageClustering, 62
* Mixture Of Gaussians
  ModelBasedClustering, 63
* MixtureOfGaussians
  ModelBasedClustering, 63
  MoGclustering, 65
* MoG
  ModelBasedClustering, 63
  MoGclustering, 65
* Model based clustering
  ModelBasedClustering, 63
* Multidimensional scaling
  ClusterPlotMDS, 28
* Neural Gas
  NeuralGasClustering, 67
* NeuralGasClustering
  NeuralGasClustering, 67
* PAM
  PAMclustering, 70
* PDE
  StatPDEdensity, 85
* PPCI
  ProjectionPursuitClustering, 74
* Pareto Density Estimation
  StatPDEdensity, 85
* Partitioning Around Medoids
  PAMclustering, 70
* Penalized Regression Based Clustering
  PenalizedRegressionBasedClustering, 72
* PenalizedRegressionBasedClustering
  PenalizedRegressionBasedClustering, 72
* Projection Based Clustering
  AutomaticProjectionBasedClustering, 10
* Projection Method
  ClusterPlotMDS, 28
* ProjectionPursuitClustering
  ProjectionPursuitClustering, 74
* Projection
  ClusterPlotMDS, 28
* QTClustering
  QTclustering, 75
* Radius
  EstimateRadiusByDistance, 42
* Rename Descending Cluster Size
  ClusterRenameDescendingSize, 31
* Robust Trimmed Clustering
  RobustTrimmedClustering, 77
* RobustTrimmedClustering
  RobustTrimmedClustering, 77
* SOM
  SOMclustering, 79
* SOTAclustering
  SOTAclustering, 81
* Self-organizing Tree Algorithm
  SOTAclustering, 81
* Shared Nearest Neighbor Clustering
  SharedNearestNeighborClustering, 78
* Spectral Clustering
  SpectralClustering, 82
  Spectrum, 83
* SpectralClustering
  SpectralClustering, 82
* Spectrum
  Spectrum, 83
* Subspace Clustering
  SubspaceClustering, 85
* SubspaceClustering
  SubspaceClustering, 85
* Tandem Clustering
  TandemClustering, 87
* TandemClustering
  TandemClustering, 87
* Target
  Target, 89
* Tetra
  Tetra, 90
* TwoDiamonds
  TwoDiamonds, 90
* WingNut
  WingNut, 91
* agnes
  AgglomerativeNestingClustering, 6
* apcluster
  APclustering, 8
* benchmarking
  FCPS-package, 3
* clara
  LargeApplicationClustering, 56
* cluster analysis
  AgglomerativeNestingClustering, 6
* clustering
  AgglomerativeNestingClustering, 6
  FCPS-package, 3
INDEX

PAMclustering, 70
* cluster
  FCPS-package, 3
* data entropy
  EntropyOfDataField, 41
* data field
  EntropyOfDataField, 41
* data set
  FCPS-package, 3
* databionic
  DBSclusteringAndVisualization, 35
* datasets
  Atom, 9
  Chainlink, 12
  EnyTime, 41
  GolfBall, 46
  Hepta, 48
  Leukemia, 58
  Lsun3D, 59
  Target, 89
  Tetra, 90
  TwoDiamonds, 90
  WingNut, 91
* density estimation
  StatPDEdensity, 85
* diana
  DivisiveAnalysisClustering, 39
* distances
  ClusterDistances, 22
  ClusterInterDistances, 24
  kmeansDist, 55
* entropy
  EntropyOfDataField, 41
* factor
  ClusterCreateClassification, 20
* fanny
  FannyClustering, 43
* fast search and find of density peaks
  ADPclustering, 5
* fuzzy clustering
  FannyClustering, 43
* ggproto density estimation
  StatPDEdensity, 85
* inter cluster
  ClusterInterDistances, 24
* intercluster
  ClusterInterDistances, 24
* intra cluster
  ClusterDistances, 22
* intracluster
  ClusterDistances, 22
* k-batch clustering
  SOMclustering, 79
* k-batch
  SOMclustering, 79
* kmeans Clustering
  kmeansClustering, 53
  kmeansDist, 55
* kmeansClustering
  kmeansClustering, 53
  kmeansDist, 55
* mst
  MSTclustering, 66
* optics
  OPTICSclustering, 68
* pdfClustering
  pdfClustering, 71
* snn
  SharedNearestNeighborClustering, 78
* som clustering
  SOMclustering, 79
* swarm
  DBSclusteringAndVisualization, 35

adpclust, 6
ADPclustering, 5, 5, 37–39
AgglomerativeNestingClustering, 6
agnes, 7
APclustering, 8
as.dendrogram, 49, 50, 61, 62
Atom, 9
AutomaticProjectionBasedClustering, 10, 74, 88

Chainlink, 12
clara, 57
ClusterabilityMDplot, 13
ClusterAccuracy(ClusteringAccuracy), 23
ClusterApply, 16
ClusterChallenge, 17
ClusterCount, 19
ClusterCreateClassification, 20
ClusterDendrogram, 21
ClusterDistances, 22
ClusteringAccuracy, 23
ClusteringAlgorithms(FCPS-package), 3
ClusterInterDistances, 23, 24
ClusterIntraDistances
   (ClusterDistances), 22
ClusterNoEstimation, 26
ClusterPlotMDS, 18, 28
ClusterRename, 30
ClusterRenameDescendingSize, 31
CrossEntropyClustering, 32
cutree, 21

DatabionicSwarmClustering
   (DBSclusteringAndVisualization), 35
DBscan, 33, 43, 86
DBSclustering, 36
DBSclusteringAndVisualization, 35
densityClust, 38, 39
DensityPeakClustering, 5, 6, 37
dist, 8
DivisiveAnalysisClustering, 39

EngyTime, 41
EntropyOfDataField, 41
EstimateRadiusByDistance, 42

FannyClustering, 43
FCPS-package, 3

GeneratePmatrix, 43
GeneratePswarmVisualization, 36
GenieClustering, 45, 51
GolfBall, 46

HCLclustering, 47
hclust, 21
Hepta, 48
Hierarchical_DBSCAN, 51, 52
HierarchicalCluster
   (HierarchicalClusterData), 48
HierarchicalClusterData, 48, 51, 52
HierarchicalClusterDists, 49, 51, 52
HierarchicalClustering, 46, 50, 62, 63

InterClusterDistances
   (ClusterInterDistances), 24
IntraClusterDistances
   (ClusterDistances), 22

kmeansClustering, 53
kmeansDist, 55

LargeApplicationClustering, 56
Leukemia, 58
Lsun3D, 59

MarkovClustering, 59
MDplot, 15, 23, 25
MinimalEnergyClustering, 51, 52, 61
MinimaxLinkageClustering, 62
ModelBasedClustering, 63, 64–66
MoGclustering, 64, 65
mst.knn, 66, 67
MSTclustering, 66

NeuralGasClustering, 67

optics, 69
OPTICSclustering, 68

PAMClustering (PAMclustering), 70
PAMclustering, 70
parDist, 38, 45, 49, 61, 62, 66
dpdfClustering, 71
PenalizedRegressionBasedClustering, 72
Plot3D, 29
plot_ly, 38
plotTopographicMap, 11
PRclust, 73
ProjectionPursuitClustering, 11, 74, 74, 88
Pswarm, 36

QTclustering (QTclustering), 75
QTclustering, 75

RobustTrimmedClustering, 77

SharedNearestNeighborClustering, 78
similarities, 8
sNNclust, 79
SOMclustering, 79
somgrid, 80
SOTAclustering, 81
sotaClustering (SOTAclustering), 81
SpectralClustering, 82
Spectrum, 83, 84
StatPDEdensity, 85
SubspaceClustering, 85

supersom, 80

TandemClustering, 11, 74, 87
INDEX

Target, 89
 TCLust, 77
 Tetra, 90
 TwoDiamonds, 90

 WingNut, 91