Package ‘densityClust’

January 29, 2024

Type Package
Title Clustering by Fast Search and Find of Density Peaks
Version 0.3.3
Maintainer Thomas Lin Pedersen <thomasp85@gmail.com>
Description An improved implementation (based on k-nearest neighbors) of the density peak clustering algorithm, originally described by Alex Rodriguez and Alessandro Laio (Science, 2014 vol. 344). It can handle large datasets (> 100,000 samples) very efficiently. It was initially implemented by Thomas Lin Pedersen, with inputs from Sean Hughes and later improved by Xiaojie Qiu to handle large datasets with kNNs.
License GPL (>= 2)
URL https://github.com/thomasp85/densityClust
BugReports https://github.com/thomasp85/densityClust/issues
Imports FNN, ggplot2, ggrepel, grDevices, gridExtra, RColorBrewer, Rtsne
Suggests covr, testthat
LinkingTo cpp11
Encoding UTF-8
RoxygenNote 7.3.1
NeedsCompilation yes
Author Thomas Lin Pedersen [aut, cre], Sean Hughes [aut], Xiaojie Qiu [aut]
Repository CRAN
Date/Publication 2024-01-29 14:30:02 UTC

R topics documented:

clustered .................................................. 2
clusters ................................................... 2
clustered  

Check whether a densityCluster object have been clustered

Description
This function checks whether `findClusters()` has been performed on the given object and returns a boolean depending on the outcome.

Usage
```
clustered(x)
```

### S3 method for class 'densityCluster'
```
clustered(x)
```

Arguments
- `x` A densityCluster object

Value
- TRUE if `findClusters()` have been performed, otherwise FALSE

clusters  

Extract cluster membership from a densityCluster object

Description
This function allows the user to extract the cluster membership of all the observations in the given densityCluster object. The output can be formatted in two ways as described below. Halo observations can be chosen to be removed from the output.

Usage
```
clusters(x, ...)
```

### S3 method for class 'densityCluster'
```
clusters(x, as.list = FALSE, halo.rm = TRUE, ...)
```
densityClust

Arguments

x

The densityClust object. findClusters() must have been performed prior to this call to avoid throwing an error.

... 

Currently ignored

as.list

Should the output be in the list format. Defaults to FALSE

halo.rm

Logical. should halo observations be removed. Defaults to TRUE

Details

Two formats for the output are available. Either a vector of integers denoting for each observation, which cluster the observation belongs to. If halo observations are removed, these are set to NA. The second format is a list with a vector for each group containing the index for the member observations in the group. If halo observations are removed their indexes are omitted. The list format correspond to the following transform of the vector format split(1:length(clusters), clusters), where clusters are the cluster information in vector format.

Value

A vector or list with cluster memberships for the observations in the initial distance matrix

densityClust

Calculate clustering attributes based on the densityClust algorithm

densityClust(distance, dc, gaussian = FALSE, verbose = FALSE, ...)

Usage

densityClust(distance, dc, gaussian = FALSE, verbose = FALSE, ...)

Arguments

distance

A distance matrix or a matrix (or data.frame) for the coordinates of the data. If a matrix or data.frame is used the distances and local density will be estimated using a fast k-nearest neighbor approach.

dc

A distance cutoff for calculating the local density. If missing it will be estimated with estimateDc(distance)
densityClust

```r

gaussian Logical. Should a gaussian kernel be used to estimate the density (defaults to FALSE)
verbose Logical. Should the running details be reported
...
Additional parameters passed on to get.knn
```

Details

The function calculates rho and delta for the observations in the provided distance matrix. If a distance cutoff is not provided this is first estimated using `estimateDc()` with default values.

The information kept in the densityCluster object is:

- **rho** A vector of local density values
- **delta** A vector of minimum distances to observations of higher density
- **distance** The initial distance matrix
- **dc** The distance cutoff used to calculate rho
- **threshold** A named vector specifying the threshold values for rho and delta used for cluster detection
- **peaks** A vector of indexes specifying the cluster center for each cluster
- **clusters** A vector of cluster affiliations for each observation. The clusters are referenced as indexes in the peaks vector
- **halo** A logical vector specifying for each observation if it is considered part of the halo
- **knn_graph** kNN graph constructed. It is only applicable to the case where coordinates are used as input. Currently it is set as NA.
- **nearest_higher_density_neighbor** index for the nearest sample with higher density. It is only applicable to the case where coordinates are used as input.
- **nn.index** indices for each cell’s k-nearest neighbors. It is only applicable for the case where coordinates are used as input.
- **nn.dist** distance to each cell’s k-nearest neighbors. It is only applicable for the case where coordinates are used as input.

Before running findClusters the threshold, peaks, clusters and halo data is NA.

Value

A densityCluster object. See details for a description.

References


See Also

`estimateDc()`, `findClusters()`
estimateDc

Examples

```r
irisDist <- dist(iris[,1:4])
irisClust <- densityClust(irisDist, gaussian=TRUE)
plot(irisClust) # Inspect clustering attributes to define thresholds

irisClust <- findClusters(irisClust, rho=2, delta=2)
plotMDS(irisClust)
split(iris[,5], irisClust$clusters)
```

---

estimateDc (Estimate the distance cutoff for a specified neighbor rate)

Description

This function calculates a distance cutoff value for a specific distance matrix that makes the average neighbor rate (number of points within the distance cutoff value) fall between the provided range. The authors of the algorithm suggests aiming for a neighbor rate between 1 and 2 percent, but also states that the algorithm is quite robust with regards to more extreme cases.

Usage

```r
estimateDc(distance, neighborRateLow = 0.01, neighborRateHigh = 0.02)
```

Arguments

- `distance`: A distance matrix
- `neighborRateLow`: The lower bound of the neighbor rate
- `neighborRateHigh`: The upper bound of the neighbor rate

Value

A numeric value giving the estimated distance cutoff value

Note

If the number of points is larger than 448 (resulting in 100,128 pairwise distances), 100,128 distance pairs will be randomly selected to speed up computation time. Use `set.seed()` prior to calling `estimateDc` in order to ensure reproducible results.

References

Examples

```r
irisDist <- dist(iris[,1:4])
estimateDc(irisDist)
```

findClusters

Detect clusters in a densityCluster object

Description

This function uses the supplied rho and delta thresholds to detect cluster peaks and assign the rest of the observations to one of these clusters. Furthermore core/halo status is calculated. If either rho or delta threshold is missing the user is presented with a decision plot where they are able to click on the plot area to set the threshold. If either rho or delta is set, this takes precedence over the value found by clicking.

Usage

```r
findClusters(x, ...)

## S3 method for class 'densityCluster'
findClusters(x, rho, delta, plot = FALSE, peaks = NULL, verbose = FALSE, ...)
```

Arguments

- `x`: A densityCluster object as produced by `densityClust()`
- `...`: Additional parameters passed on
- `rho`: The threshold for local density when detecting cluster peaks
- `delta`: The threshold for minimum distance to higher density when detecting cluster peaks
- `plot`: Logical. Should a decision plot be shown after cluster detection
- `peaks`: A numeric vector indicates the index of density peaks used for clustering. This vector should be retrieved from the decision plot with caution. No checking involved.
- `verbose`: Logical. Should the running details be reported

Value

A densityCluster object with clusters assigned to all observations

References

Examples

```
irisDist <- dist(iris[,1:4])
irisClust <- densityClust(irisDist, gaussian=TRUE)
plot(irisClust) # Inspect clustering attributes to define thresholds

irisClust <- findClusters(irisClust, rho=2, delta=2)
plotMDS(irisClust)
split(iris[,5], irisClust$clusters)
```

---

**plotDensityClust**  
*Plot densityClust results*

**Description**

Generate a single panel of up to three diagnostic plots for a densityClust object.

**Usage**

```
plotDensityClust(
  x, 
  type = "all", 
  n = 20, 
  mds = NULL, 
  dim.x = 1, 
  dim.y = 2, 
  col = NULL, 
  alpha = 0.8
)
```

**Arguments**

- **x**: A densityCluster object as produced by `densityClust`
- **type**: A character vector designating which figures to produce. Valid options include "dg" for a decision graph of $\delta$ vs. $\rho$, "gg" for a gamma graph depicting the decrease of $\gamma = \delta \times \rho$ across samples, and "mds", for a Multi-Dimensional Scaling (MDS) plot of observations. Any combination of these three can be included in the vector, or to produce all plots, specify `type = "all"`.
- **n**: Number of observations to plot in the gamma graph.
- **mds**: A matrix of scores for observations from a Principal Components Analysis or MDS. If omitted, and a MDS plot has been requested, one will be calculated.
- **dim.x**, **dim.y**: The numbers of the dimensions to plot on the x and y axes of the MDS plot.
- **col**: Vector of colors for clusters.
- **alpha**: Value in $0:1$ controlling transparency of points in the decision graph and MDS plot.
Description

This function produces an MDS scatterplot based on the distance matrix of the densityCluster object (if there is only the coordinates information, a distance matrix will be calculated first), and, if clusters are defined, colours each observation according to cluster affiliation. Observations belonging to a cluster core is plotted with filled circles and observations belonging to the halo with hollow circles. This plotting is not suitable for running large datasets (for example datasets with > 1000 samples). Users are suggested to use other methods, for example tSNE, etc. to visualize their clustering results too.
Usage

plotMDS(x, ...)

Arguments

x A densityCluster object as produced by densityClust()
...

Additional parameters. Currently ignored

See Also

densityClust() for creating densityCluster objects, and plotTSNE() for an alternative plotting approach.

Examples

irisDist <- dist(iris[,1:4])
irisClust <- densityClust(irisDist, gaussian=TRUE)
plot(irisClust) # Inspect clustering attributes to define thresholds

irisClust <- findClusters(irisClust, rho=2, delta=2)
plotMDS(irisClust)
split(iris[,5], irisClust$clusters)

\[
\text{plotTSNE} \quad \text{Plot observations using t-distributed neighbor embedding and colour by cluster}
\]

Description

This function produces an t-SNE scatterplot based on the distance matrix of the densityCluster object (if there is only the coordinates information, a distance matrix will be calculate first), and, if clusters are defined, colours each observation according to cluster affiliation. Observations belonging to a cluster core is plotted with filled circles and observations belonging to the halo with hollow circles.

Usage

plotTSNE(x, ...)

Arguments

x A densityCluster object as produced by densityClust()
...

Additional parameters. Currently ignored
See Also

densityClust() for creating densityCluster objects, and plotMDS() for an alternative plotting approach.

Examples

irisDist <- dist(iris[,1:4])
irisClust <- densityClust(irisDist, gaussian=TRUE)
plot(irisClust) # Inspect clustering attributes to define thresholds

irisClust <- findClusters(irisClust, rho=2, delta=2)
plotTSNE(irisClust)
split(iris[,5], irisClust$clusters)
Index

classified, 2
clusters, 2
densityClust, 3, 7
densityClust(), 6, 9, 10
estimateDc, 5
estimateDc(), 4
findClusters, 6
findClusters(), 2–4
get.knn, 4
plotDensityClust, 7
plotMDS, 8
plotMDS(), 10
plotTSNE, 9
plotTSNE(), 9
set.seed(), 5