Package 'genie'

August 3, 2020

Type Package

Title Fast, Robust, and Outlier Resistant Hierarchical Clustering

Version 1.0.5

Date 2020-08-02

Description Includes the reference implementation of Genie - a hierarchical clustering algorithm that links two point groups in such a way that an inequity measure (namely, the Gini index) of the cluster sizes does not significantly increase above a given threshold.
This method most often outperforms many other data segmentation approaches in terms of clustering quality as tested on a wide range of benchmark datasets. At the same time, Genie retains the high speed of the single linkage approach, therefore it is also suitable for analysing larger data sets.
For more details see (Gagolewski et al. 2016 <DOI:10.1016/j.ins.2016.05.003>).
For an even faster and more feature-rich implementation, including, amongst others, noise point detection, see the 'genieclust' package.

License GPL (>= 3)

BugReports http://github.com/gagolews/genie/issues

URL http://genieclust.gagolewski.com/

Depends R (>= 3.3.0), stats, genieclust

Imports Rcpp (>= 1.0.0)

Suggests datasets, testthat, stringi

LinkingTo Rcpp (>= 1.0.0)

SystemRequirements OpenMP, C++11

RoxygenNote 7.1.1

NeedsCompilation yes

Author Marek Gagolewski [aut, cre, cph]
(<https://orcid.org/0000-0003-0637-6028>),
Maciej Bartoszuk [aut] (<https://orcid.org/0000-0001-6088-8273>),
Anna Cena [aut] (<https://orcid.org/0000-0001-8697-5383>)

Maintainer Marek Gagolewski <marek@gagolewski.com>

Repository CRAN

Date/Publication 2020-08-02 22:00:02 UTC

R topics documented:

`genie-package` *The Genie Package*

Description

See `hclust2()` for details.

Author(s)

Marek Gagolewski, Maciej Bartoszuk, Anna Cena

`hclust2` *Fast Hierarchical Clustering in Spaces Equipped With a Dissimilarity Measure*

Description

The reference implementation of the fast, robust and outlier resistant Genie algorithm described in (Gagolewski, Bartoszuk, Cena, 2016). Note that the `genie` package has been superseded by `genieclust`, see `gclust` and `genie` for more details.

Usage

```
hclust2(d = NULL, objects = NULL, thresholdGini = 0.3, useVpTree = FALSE, ...)
```

Arguments

`d`	an object of class `dist`, `NULL`, or a single string, see below
`objects`	`NULL`, numeric matrix, a list, or a character vector
`thresholdGini`	single numeric value in [0,1], threshold for the Gini index, 1 gives the standard single linkage algorithm
`useVpTree`	single logical value, whether to use a vantage-point tree to speed up nearest neighbour searching in low-dimensional spaces
`...`	internal parameters used to tune up the algorithm

Details

The time needed to apply a hierarchical clustering algorithm is most often dominated by the number of computations of a pairwise dissimilarity measure. Such a constraint, for larger data sets, puts at a disadvantage the use of all the classical linkage criteria but the single linkage one. However, it is known that the single linkage clustering algorithm is very sensitive to outliers, produces highly skewed dendrograms, and therefore usually does not reflect the true underlying data structure – unless the clusters are well-separated.

To overcome its limitations, in (Gagolewski, Bartoszuk, Cena, 2016) we proposed a new hierarchical clustering linkage criterion. Namely, our algorithm links two clusters in such a way that a chosen economic inequity measure (here, the Gini index) of the cluster sizes does not increase drastically above a given threshold. The benchmarks indicate a high practical usefulness of the introduced method: it most often outperforms the Ward or average linkage in terms of the clustering quality while retaining the single linkage speed. The algorithm can be run in parallel (via OpenMP) on multiple threads to speed up its execution further on. Its memory overhead is small: there is no need to precompute the complete distance matrix to perform the computations in order to obtain a desired clustering.

For compatibility with `hclust`, `d` may be an object of class `dist`. In such a case, the `objects` argument is ignored. Note that such an object requires ca. *8n(n-1)/2* bytes of computer's memory, where *n* is the number of objects to cluster, and therefore this setting can be used to analyse data sets of sizes up to about 10,000-50,000.

If `objects` is a character vector or a list, then `d` should be a single string, one of: `levenshtein` (or `NULL`), `hamming`, `dinu` (Dinu, Sgarro, 2006), or `euclinf` (Cena et al., 2015). Note that the list must consist either of integer or of numeric vectors only (depending on the dissimilarity measure of choice). On the other hand, each string must be in ASCII, but you can always convert it to UTF-32 with `stri_enc_toutf32`.

Otherwise, if `objects` is a numeric matrix (here, each row denotes a distinct observation), then `d` should be a single string, one of: `euclidean_squared` (or `NULL`), `euclidean` (which yields the same results as `euclidean_squared`) `manhattan`, `maximum`, or `hamming`.

If `useVpTree` is `FALSE`, then the dissimilarity measure of choice is guaranteed to be computed for each unique pair of `objects` only once.

Value

A named list of class `hclust`, see `hclust`, with additional components:

- `stats` - performance statistics
- `control` - internal parameters used

References

Cena A., Gagolewski M., Mesiar R., Problems and challenges of information resources producers' clustering, *Journal of Informetrics* 9(2), 2015, pp. 273-284.

Dinu L.P., Sgarro A., A Low-complexity Distance for DNA Strings, *Fundamenta Informaticae* 73(3), 2006, pp. 361-372.

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

Gagolewski M., Cena A., Bartoszuk M. *Hierarchical clustering via penalty-based aggregation and the Genie approach*, In: Torra V. et al. (Eds.), *Modeling Decisions for Artificial Intelligence* (*Lecture Notes in Artificial Intelligence* 9880), Springer, 2016.

Examples

```
library("datasets")
data("iris")
h <- hclust2(objects=as.matrix(iris[,2:3]), thresholdGini=0.2)
plot(iris[,2], iris[,3], col=cutree(h, 3), pch=as.integer(iris[,5]), asp=1, las=1)
```

Index