agnes()

#R #StatisticalModeling #cluster

the following is from the help feature in RStudio

Agglomerative Nesting (Hierarchical Clustering)

Description

Computes agglomerative hierarchical clustering of the dataset.

Usage

```
agnes(x, diss = inherits(x, "dist"), metric = "euclidean",
      stand = FALSE, method = "average", par.method,
      keep.diss = n < 100, keep.data = !diss, trace.lev = 0)
```

Arguments

- x
  - data matrix or data frame, or dissimilarity matrix, depending on the value of the diss argument.
  - In case of a matrix or data frame, each row corresponds to an observation, and each column corresponds to a variable. All variables must be numeric. Missing values (NAs) are allowed.
  - In case of a dissimilarity matrix, x is typically the output of daisy or dist. Also a vector with length n*(n-1)/2 is allowed (where n is the number of observations), and will be interpreted in the same way as the output of the above-mentioned functions. Missing values (NAs) are not allowed.
- diss
  - logical flag: if TRUE (default for dist or dissimilarity objects), then x is assumed to be a dissimilarity matrix. If FALSE, then x is treated as a matrix of observations by variables.
- metric
  - character string specifying the metric to be used for calculating dissimilarities between observations. The currently available options are "euclidean" and "manhattan". Euclidean distances are root sum-of-squares of differences, and manhattan distances are the sum of absolute differences. If x is already a dissimilarity matrix, then this argument will be ignored.
- stand
  - logical flag: if TRUE, then the measurements in x are 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 x is already a dissimilarity matrix, then this argument will be ignored.
- method

    - character string defining the clustering method. The six methods implemented are "average" (unweighted pair-group arithMetic average method, aka ‘UPGMA’), "single" (single linkage), "complete" (complete linkage), "ward" (Ward's method), "weighted" (weighted average linkage, aka ‘WPGMA’), its generalization "flexible" which uses (a constant version of) the Lance-Williams formula and the par.method argument, and "gaverage" a generalized "average" aka “flexible UPGMA” method also using the Lance-Williams formula and par.method.
    - The default is "average".
- par.method
    - If method is "flexible" or "gaverage", a numeric vector of length 1, 3, or 4, (with a default for "gaverage"), see in the details section.
- keep.diss, keep.data
    - logicals indicating if the dissimilarities and/or input data x should be kept in the result. Setting these to FALSE can give much smaller results and hence even save memory allocation time.
- trace.lev
    - integer specifying a trace level for printing diagnostics during the algorithm. Default 0 does not print anything; higher values print increasingly more.

Details
agnes is fully described in chapter 5 of Kaufman and Rousseeuw (1990). Compared to other agglomerative clustering methods such as hclust, agnes has the following features: (a) it yields the agglomerative coefficient (see agnes.object) which measures the amount of clustering structure found; and (b) apart from the usual tree it also provides the banner, a novel graphical display (see plot.agnes).

The agnes-algorithm constructs a hierarchy of clusterings.
At first, each observation is a small cluster by itself. Clusters are merged until only one large cluster remains which contains all the observations. At each stage the two nearest clusters are combined to form one larger cluster.

For method="average", the distance between two clusters is the average of the dissimilarities between the points in one cluster and the points in the other cluster.
In method="single", we use the smallest dissimilarity between a point in the first cluster and a point in the second cluster (nearest neighbor method).
When method="complete", we use the largest dissimilarity between a point in the first cluster and a point in the second cluster (furthest neighbor method).

The method = "flexible" allows (and requires) more details: The Lance-Williams formula specifies how dissimilarities are computed when clusters are agglomerated (equation (32) in K&R(1990), p.237). If clusters C_1 and C_2 are agglomerated into a new cluster, the dissimilarity between their union and another cluster Q is given by

$$D(C_1 \cup C_2, Q) = \alpha_1 * D(C_1, Q) + \alpha_2 * D(C_2, Q) + \beta * D(C_1, C_2) + \gamma * |D(C_1, Q) - D(C_2, Q)|,$$

where the four coefficients (α_1, α_2, β, γ) are specified by the vector par.method, either directly as vector of length 4, or (more conveniently) if par.method is of length 1, say = α, par.method is extended to give the “Flexible Strategy” (K&R(1990), p.236 f) with Lance-Williams coefficients (α_1 = α_2 = α, β =

1 - 2α, γ=0).
Also, if length(par.method) == 3, γ = 0 is set.

Care and expertise is probably needed when using method = "flexible" particularly for the case when par.method is specified of longer length than one. Since cluster version 2.0, choices leading to invalid merge structures now signal an error (from the C code already). The weighted average (method="weighted") is the same as method="flexible", par.method = 0.5. Further, method= "single" is equivalent to method="flexible", par.method = c(.5,.5,0,-.5), and method="complete" is equivalent to method="flexible", par.method = c(.5,.5,0,+.5).

The method = "gaverage" is a generalization of "average", aka “flexible UPGMA” method, and is (a generalization of the approach) detailed in Belbin et al. (1992). As "flexible", it uses the Lance-Williams formula above for dissimilarity updating, but with α_1 and α_2 not constant, but proportional to the sizes n_1 and n_2 of the clusters C_1 and C_2 respectively, i.e,

$$\alpha_j = \alpha'_j * n_1/(n_1 + n_2),$$

where α'_1, α'_2 are determined from par.method, either directly as (α_1, α_2, β, γ) or (α_1, α_2, β) with γ = 0, or (less flexibly, but more conveniently) as follows:

Belbin et al proposed “flexible beta”, i.e. the user would only specify β (as par.method), sensibly in

$$-1 \le \beta < 1,$$

and β determines α'_1 and α'_2 as

$$\alpha'_j = 1 - \beta,$$

and γ = 0.

This β may be specified by par.method (as length 1 vector), and if par.method is not specified, a default value of -0.1 is used, as Belbin et al recommend taking a β value around -0.1 as a general agglomerative hierarchical clustering strategy.

Note that method = "gaverage", par.method = 0 (or par.method = c(1,1,0,0)) is equivalent to the agnes() default method "average".

Value
an object of class "agnes" (which extends "twins") representing the clustering. See agnes.object for details, and methods applicable.

BACKGROUND
Cluster analysis divides a dataset into groups (clusters) of observations that are similar to each other.

Hierarchical methods
like agnes, diana, and mona construct a hierarchy of clusterings, with the number of clusters ranging from one to the number of observations.

Partitioning methods
like pam, clara, and fanny require that the number of clusters be given by the user.

Examples

```
data(votes.repub)
agn1 <- agnes(votes.repub, metric = "manhattan", stand = TRUE)
agn1
plot(agn1)

op <- par(mfrow=c(2,2))
agn2 <- agnes(daisy(votes.repub), diss = TRUE, method = "complete")
plot(agn2)
## alpha = 0.625 ==> beta = -1/4  is "recommended" by some
agnS <- agnes(votes.repub, method = "flexible", par.meth = 0.625)
plot(agnS)
par(op)

## "show" equivalence of three "flexible" special cases
d.vr <- daisy(votes.repub)
a.wgt  <- agnes(d.vr, method = "weighted")
a.sing <- agnes(d.vr, method = "single")
a.comp <- agnes(d.vr, method = "complete")
iC <- -(6:7) # not using 'call' and 'method' for comparisons
stopifnot(
  all.equal(a.wgt [iC], agnes(d.vr, method="flexible", par.method = 0.5)[iC])   ,
  all.equal(a.sing[iC], agnes(d.vr, method="flex", par.method= c(.5,.5,0, -.5))
[iC]),
  all.equal(a.comp[iC], agnes(d.vr, method="flex", par.method= c(.5,.5,0, +.5))
[iC]))

## Exploring the dendrogram structure
(d2 <- as.dendrogram(agn2)) # two main branches
d2[[1]] # the first branch
d2[[2]] # the 2nd one  { 8 + 42  = 50 }
d2[[1]][[1]]# first sub-branch of branch 1 .. and shorter form
identical(d2[[c(1,1)]],
          d2[[1]][[1]])
## a "textual picture" of the dendrogram :
str(d2)

data(agriculture)

## Plot similar to Figure 7 in ref
## Not run: plot(agnes(agriculture), ask = TRUE)



data(animals)
aa.a  <- agnes(animals) # default method = "average"
```

```
aa.ga <- agnes(animals, method = "gaverage")
op <- par(mfcol=1:2, mgp=c(1.5, 0.6, 0), mar=c(.1+ c(4,3,2,1)),
          cex.main=0.8)
plot(aa.a,  which.plot = 2)
plot(aa.ga, which.plot = 2)
par(op)


## Show how "gaverage" is a "generalized average":
aa.ga.0 <- agnes(animals, method = "gaverage", par.method = 0)
stopifnot(all.equal(aa.ga.0[iC], aa.a[iC]))
```