Cluster Analysis: nagdmc_hclust

Purpose

nagdmc_hclust computes an hierarchical cluster analysis.

Declaration

```
#include <nagdmc.h>
void nagdmc_hclust(long rec1, long nvar, long nrec, long dblk, double data[],
                   long nxvar, long xvar[], int ctype, long lmerge[],
                   long umerge[], double mergedist[], long denord[],
                   double dendist[], int *info);
```

Parameters

1: **rec1** – `long` *Input*

On entry: the index in the data of the first data record used in the analysis.

Constraint: **rec1** ≥ 0.

2: **nvar** – `long` *Input*

On entry: the number of variables in the data.

Constraint: **nvar** ≥ 1.

3: **nrec** – `long` *Input*

On entry: the number of consecutive records, beginning at **rec1**, used in the analysis.

Constraint: **nrec** > 1.

4: **dblk** – `long` *Input*

On entry: the total number of records in the data block.

Constraint: **dblk** $\geq$ **rec1** + **nrec**.

5: **data**[**dblk** $*$ **nvar**] – `double` *Input*

On entry: the data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar**-1) are stored in **data**[$i*$**nvar**$+j$], for $i = 0, 1, \ldots,$ **dblk** $- 1$.

6: **nxvar** – `long` *Input*

On entry: the number of variables in the analysis. If **nxvar** $= 0$, all variables in the data are used in the analysis.

Constraint: $0 \leq$ **nxvar** $\leq$ **nvar**.

7: **xvar**[**nxvar**] – `long` *Input*

On entry: the indices indicating the position in **data** in which the variables are stored. If **nxvar** $= 0$ then **xvar** must be 0, and the indices of variables are given by $j = 0, 1, \ldots,$ **nvar** $- 1$.

Constraints: if **nxvar** > 0, $0 \leq$ **xvar**$[i] <$ **nvar**, for $i = 0, 1, \ldots,$ **nxvar** $- 1$; otherwise **xvar** must be 0.

8: **ctype** – `int` *Input*

On entry: if **ctype** is set equal to zero, the group average method is used; otherwise **ctype** must equal one, and the minimum variance method is used.

Constraint: **ctype** $\in \{0, 1\}$.

9: **lmerge**[**nrec**-1] – `double` *Output*

On exit: see **mergedist**.

10: **umerge**[**nrec**-1] – `double` *Output*

On exit: see **mergedist**.

11: **mergedist**[**nrec**−1] – `double` *Output*

On exit: cluster **umerge**[i] is merged into cluster **lmerge**[i] at a distance **mergedist**[i], for $i = 0, 1, \ldots,$ **nrec** $- 2$.

12: **denord**[**nrec**] – `long` *Output*

On exit: the order of the data records required for producing a dendrogram.

13: **dendist**[**nrec**] – `double` *Output*

On exit: **dendist**[i] is the distance at which cluster **denord**[i] merges with cluster **denord**[$i + 1$]. **dendist**[**nrec** $- 1$] contains the maximum distance.

14: **info** – `int *` *Output*

On exit: **info** gives information on the success of the function call:

0: the function successfully completed its task.

i; $i = 1, 2, 3, 4, 6, 7, 8$: the specification of the ith formal parameter was incorrect.

99: the function failed to allocate enough memory.

100: an internal error occurred during the execution of the function.

Notation

nrec	the number of data records, n.
nxvar	the number of variables, p.
lmerge[i]	j at step $i + 1$ of the clustering.
umerge[i]	k at step $i + 1$ of the clustering.
mergedist[i]	the distance d_{jk} at which clusters j and k merged at step $i + 1$ of the clustering.
denord	the order, $\mathcal{R}$, of data records for a dendrogram.
dendist	the merged distances in the order required by **denord**.

Description

Given n data records, a distance or dissimilarity matrix is a symmetric matrix with zero diagonal elements such that the jkth element represents how far apart or how dissimilar the jth and kth data records are.

Let X be an n by p data matrix of observations of p variables on n data records, then the squared Euclidean distance between the jth and kth data records, d_{jk}, is:

$$d_{jk} = \sum_{i=1}^{p} (x_{ji} - x_{ki})^2,$$

where x_{ji} and x_{ki} are the (j, i)th and (k, i)th elements of X.

Given a distance or dissimilarity matrix for n data records, cluster analysis aims to group the n data records into a number of similar groups or clusters. With agglomerative clustering methods, an hierarchical tree is produced by starting with n clusters, each containing a single record. At each of the $n - 1$ stages, the method merges the two nearest clusters to form a larger cluster. This process continues until all data records belong to a single cluster.

Methods differ as to how the distance between the new cluster and other clusters are computed. For three clusters i, j and k let n_i, n_j and n_k be the number of data records in each cluster and let d_{ij}, d_{ik} and d_{jk} be the distances between the clusters. Let clusters j and k be merged to give cluster jk, then the distance from cluster i to cluster jk, $d_{i.jk}$, can be computed in the following ways.

(a) Group average: $d_{i.jk} = \dfrac{n_j}{n_j + n_k} d_{ij} + \dfrac{n_k}{n_j + n_k} d_{ik}$;

(b) Minimum variance: $d_{i.jk} = \left[(n_i + n_j)d_{ij} + (n_i + n_k)d_{ik} - n_i d_{jk}\right] / (n_i + n_j + n_k)$.

For convenience, clusters are numbered $0, 1, \ldots, n - 1$. If clusters j and k, $j < k$, merge then the new cluster will be referred to as cluster j. Information on the merging of clusters is given by the

values of j, k and d_{jk} for each of the $n-1$ clustering steps. The results of an hierarchical clustering can be represented as a tree diagram, also known as a dendrogram, as shown in Figure 1. The order of data records, $\mathcal{R}$, required to draw a dendrogram without crossing dendrites is returned by the function. This ordering is computed so that the first element is 0. The distances associated with this ordering are also returned by the function.

In Figure 1 the hierarchical clustering of five data records proceeds as follows. Reading the dendrogram from the record labels upwards, each record begins in its own cluster giving five clusters labelled $0, 1, \ldots, .4$. At a distance of 2, clusters 0 and 1 combine to give four clusters labelled 0, 2, 3 and 4. Similarly, at a distance of 4 clusters 2 and 3 combine giving three clusters labelled 0, 2 and 4. At distance 9, cluster 4 joins the cluster labelled 2, giving two clusters labelled 0 and 2. Finally, at a distance of 11 clusters 0 and 2 join to give a single cluster labelled 0.

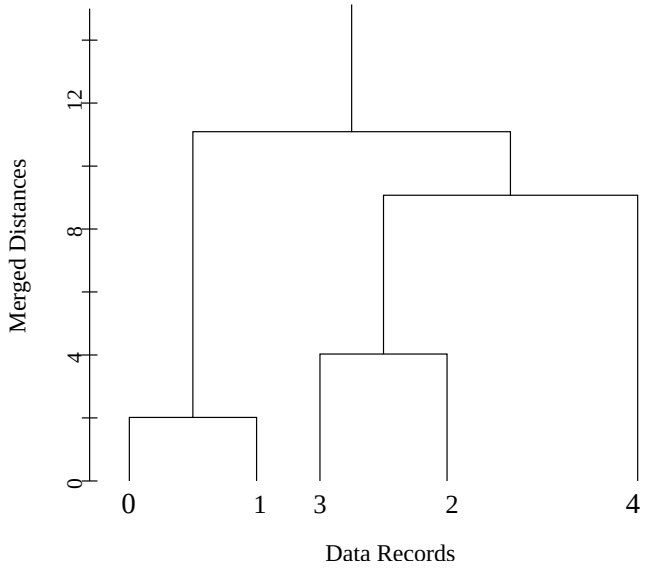

Figure 1: *Dendrogram of example results from an hierarchical clustering. The order of data record labels, $\mathcal{R}$, ensures that none of the dendrites cross in the dendrogram.*

This implementation of an hierarchical clustering procedure does not store in memory the dissimilarity matrix, thus reducing the storage required by $\mathcal{O}(n^2)$. Hence this method can be applied to higher numbers of data records than standard implementations.

References and Further Reading

Everitt B S (1974) *Cluster Analysis* Heinemann.

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press.

See Also

nagdmc_cind	computes cluster memberships following an hierarchical clustering.
nagdmc_tab2	cross-tabulates known groupings and cluster memberships.
hclust_ex.c	the example calling program.