Least Squares Fit of Ultrametrics to Dissimilarities

Description

Find the ultrametric with minimal square distance (Euclidean dissimilarity) to given dissimilarity objects.

Usage

```
ls_fit_ultrametric(x, method = c("SUMT", "IP", "IR"), weights = 1,
                   control = list())
```

Arguments

`x`	a dissimilarity object inheriting from or coercible to class "`dist`", or an ensemble of such objects.
`method`	a character string indicating the fitting method to be employed. Must be one of "`SUMT`"(default), "`IP`", or "`IR`", or a unique abbreviation thereof.
`weights`	a numeric vector or matrix with non-negative weights for obtaining a weighted least squares fit. If a matrix, its numbers of rows and columns must be the same as the number of objects in `x`, and the lower diagonal part is used. Otherwise, it is recycled to the number of elements in `x`.
`control`	a list of control parameters. See **Details**.

Details

For a single dissimilarity object `x`, the problem to be solved is minimizing

$$L(u) = \sum_{i,j} w_{ij} (x_{ij} - u_{ij})^2$$

over all u satisfying the ultrametric constraints (i.e., for all i, j, k, $u_{ij} \le \max(u_{ik}, u_{jk})$). This problem is known to be NP hard (Krivanek and Moravek, 1986).

For an ensemble of dissimilarity objects, the criterion function is

$$L(u) = \sum_b w_b \sum_{i,j} w_{ij} (x_{ij}(b) - u_{ij})^2,$$

where w_b is the weight given to element x_b of the ensemble and can be specified via control parameter `weights` (default: all ones). This problem reduces to the above basic problem with x as thew_b-weighted mean of the x_b.

We provide three heuristics for solving the basic problem.

Method "SUMT" implements the SUMT (Sequential Unconstrained Minimization Technique, Fiacco and McCormick, 1968) approach of de Soete (1986) which in turn simplifies the suggestions in Carroll and Pruzansky (1980). (See sumt for more information on the SUMT approach.) We then use a final single linkage hierarchical clustering step to ensure that the returned object exactly satisfies the ultrametric constraints. The starting value *u_0* is obtained by "random shaking" of the given dissimilarity object (if not given). If there are missing values in x, i.e., the given dissimilarities are *incomplete*, we follow a suggestion of de Soete (1984), imputing the missing values by the weighted mean of the non-missing ones, and setting the corresponding weights to zero.

Available control parameters are method, control, eps, q, and verbose, which have the same roles as for sumt, and the following.

nruns

an integer giving the number of runs to be performed. Defaults to 1.

start

a single dissimilarity, or a list of dissimilarities to be employed as starting values.

The default optimization using conjugate gradients should work reasonably well for medium to large size problems. For "small" ones, using nlm is usually faster. Note that the number of ultrametric constraints is of the order *n^3*, where *n* is the number of objects in the dissimilarity object, suggesting to use the SUMT approach in favor of constrOptim.

If starting values for the SUMT are provided via start, the number of starting values gives the number of runs to be performed, and control option nruns is ignored. Otherwise, nruns starting values are obtained by random shaking of the dissimilarity to be fitted. In the case of multiple SUMT runs, the (first) best solution found is returned.

Method "IP" implements the Iterative Projection approach of Hubert and Arabie (1995). This iteratively projects the current dissimilarities to the closed convex set given by the ultrametric constraints (3-point conditions) for a single index triple *(i, j, k)*, in fact replacing the two largest values among *d_{ij}, d_{ik}, d_{jk}* by their mean. The following control parameters can be provided via the control argument.

nruns

an integer giving the number of runs to be performed. Defaults to 1.

order

a permutation of the numbers from 1 to the number of objects in x, specifying the order in which the ultrametric constraints are considered, or a list of such permutations.

maxiter

an integer giving the maximal number of iterations to be employed.

tol

a double indicating the maximal convergence tolerance. The algorithm stops if the total absolute change in the dissimilarities in an iteration is less than `tol`.

`verbose`

a logical indicating whether to provide some output on minimization progress. Defaults to`getOption("verbose")`.

If permutations are provided via `order`, the number of these gives the number of runs to be performed, and control option `nruns` is ignored. Otherwise, `nruns` randomly generated orders are tried. In the case of multiple runs, the (first) best solution found is returned.

Non-identical weights and incomplete dissimilarities are currently not supported.

Method `"IR"` implements the Iterative Reduction approach suggested by Roux (1988), see also Barthélémy and Guénoche (1991). This is similar to the Iterative Projection method, but modifies the dissimilarities between objects proportionally to the aggregated change incurred from the ultrametric projections. Available control parameters are identical to those of method `"IP"`.

Non-identical weights and incomplete dissimilarities are currently not supported.

It should be noted that all methods are heuristics which can not be guaranteed to find the global minimum. Standard practice would recommend to use the best solution found in “sufficiently many” replications of the base algorithm.

Value

An object of class `"cl_ultrametric"` containing the fitted ultrametric distances.

References

J.-P. Barthélémy and A. Guénoche (1991). *Trees and proximity representations*. Chichester: John Wiley & Sons. ISBN 0-471-92263-3.

J. D. Carroll and S. Pruzansky (1980). Discrete and hybrid scaling models. In E. D. Lantermann and H. Feger (eds.), *Similarity and Choice*. Bern (Switzerland): Huber.

L. Hubert and P. Arabie (1995). Iterative projection strategies for the least squares fitting of tree structures to proximity data. *British Journal of Mathematical and Statistical Psychology*, **48**, 281–317. doi: 10.1111/j.2044-8317.1995.tb01065.x.

M. Krivanek and J. Moravek (1986). NP-hard problems in hierarchical tree clustering. *Acta Informatica*,**23**, 311–323. doi: 10.1007/BF00289116.

M. Roux (1988). Techniques of approximation for building two tree structures. In C. Hayashi and E. Diday and M. Jambu and N. Ohsumi (Eds.), *Recent Developments in Clustering and Data Analysis*, pages 151–170. New York: Academic Press.

G. de Soete (1984). Ultrametric tree representations of incomplete dissimilarity data. *Journal of Classification*, **1**, 235–242. doi: 10.1007/BF01890124.

G. de Soete (1986). A least squares algorithm for fitting an ultrametric tree to a dissimilarity matrix.*Pattern Recognition Letters*, **2**, 133–137. doi: 10.1016/0167-8655(84)90036-9.

See Also

`cl_consensus` for computing least squares (Euclidean) consensus hierarchies by least squares fitting of average ultrametric distances; `l1_fit_ultrametric`.

Examples

```
## Least squares fit of an ultrametric to the Miller-Nicely consonant
## phoneme confusion data.
data("Phonemes")
## Note that the Phonemes data set has the consonant misclassification
## probabilities, i.e., the similarities between the phonemes.
d <- as.dist(1 - Phonemes)
u <- ls_fit_ultrametric(d, control = list(verbose = TRUE))
## Cophenetic correlation:
cor(d, u)
## Plot:
plot(u)
## ("Basically" the same as Figure 1 in de Soete (1986).)
```