Package 'nomclust'

July 2, 2020

Title Hierarchical Cluster Analysis of Nominal Data

Author Zdenek Sulc [aut, cre],
Jana Cibulkova [aut],
Hana Rezankova [aut]

Maintainer Zdenek Sulc <zdenek.sulc@vse.cz>

Version 2.1.6

Date 2020-06-17

Description Similarity measures for hierarchical clustering of objects characterized by nominal (categorical) variables. Evaluation criteria for nominal data clustering.

Depends cluster, plyr, methods

License GPL (>= 2)

RoxygenNote 7.0.2

NeedsCompilation no

Encoding UTF-8

Repository CRAN

Date/Publication 2020-07-02 08:40:02 UTC

R topics documented:

CA.methods 2
data20 2
dend.plot 3
eskin 4
eval.plot 6
evalclust 7
good1 9
good2 10
good3 11
good4 12
iof 13
lin 14
lin1 15

morlini 17
nomclust 18
nomprox 21
of 22
sm 23
ve 24
vm 25

Index **27**

`CA.methods` *Selected clustering algorithms*

Description

The dataset contains five different characteristics of 24 clustering algorithms. The "Type" variable expresses the principle on which the clustering is based. There are five possible categories: density, grid, hierarchical, model-based, and partitioning. The binary variable "OptClu" indicates if the clustering algorithm offers the optimal number of clusters. The variable "Large" indicates if the clustering algorithm was designed to cluster large datasets. The "TypicalType" variable presents the typical data type for which the clustering algorithm was determined. There are three possible categories: categorical, mixed, and quantitative. Since some clustering algorithms support more data types, the binary variable "MoreTypes" indicates this support.

Usage

```
data("CA.methods")
```

Format

A data frame containing 5 variables and 24 cases.

Source

created by the authors of the nomclust package

`data20` *Artificial nominal dataset*

Description

This dataset consists of 5 nominal variables and 20 cases. Its main aim is to demonstrate the desired entry data structure for the nomclust package.

Usage

```
data(data20)
```

Format

A data frame containing 5 variables and 20 cases.

Source

created by the authors of the nomclust package

`dend.plot` *Visualization of Cluster Hierarchy using a Dendrogram*

Description

The function `dend.plot()` visualizes the hierarchy of clusters using a dendrogram. The function also enables a user to mark the individual clusters with colors. The number of displayed clusters can be defined either by a user or by one of the five evaluation criteria.

Usage

```
dend.plot(
  x,
  clusters = "BIC",
  style = "greys",
  colorful = TRUE,
  clu.col = NA,
  main = "Dendrogram",
  ac = TRUE,
  ...
)
```

Arguments

`x`	An output of the `nomclust()` or `nomprox()` functions containing the `dend` component.
`clusters`	Either a *numeric* value or a *character* string with the name of the evaluation criterion expressing the number of displayed clusters in a dendrogram. The following evaluation criteria can be used: `"AIC"`, `"BIC"`, `"BK"`, `"PSFE"` and `"PSFM"`.
`style`	A *character* string or a *vector* of colors defines a graphical style of the produced plots. There are two predefined styles in the **nomclust** package, namely `"greys"` and `"dark"`, but a custom color scheme can be set by a user as a vector of a length four.
`colorful`	A *logical* argument specifying if the output will be colorful or black and white.
`clu.col`	An optional *vector* of colors which allows a researcher to apply user-defined colors for displayed (marked) clusters in a dendrogram.
`main`	A *character* string with the chart title.
`ac`	A *logical* argument indicating if an agglomerative coefficient will be present in the output.
`...`	Other graphical arguments compatible with the generic `plot()` function.

Details

The function can be applied to a `nomclust()` or `nomprox()` output containing the `dend` component. This component is not available when the optimization process is used.

Value

The function returns a dendrogram describing the hierarchy of clusters that can help to identify the optimal number of clusters.

Author(s)

Jana Cibulkova and Zdenek Sulc.
Contact: <jana.cibulkova@vse.cz>

See Also

`eval.plot`, `nomclust`, `nomprox`.

Examples

```
# sample data
data(data20)

# creating an object with results of hierarchical clustering
hca.object <- nomclust(data20, measure = "iof", eval = TRUE, opt = FALSE)

# a basic plot
dend.plot(hca.object)

# a dendrogram with color-coded clusters according to the BIC index
dend.plot(hca.object, clusters = "BIC", colorful = TRUE)

# using a dark style and specifying own colors in a solution with three clusters
dend.plot(hca.object, clusters = 3, style = "dark", clu.col = c("blue", "red", "green"))

# a black and white dendrogram
dend.plot(hca.object, clusters = 3, style = "dark", colorful = FALSE)
```

`eskin` *Eskin (ES) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the ES similarity measure.

Usage

```
eskin(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Eskin similarity measure was proposed by Eskin et al. (2002) and examined by Boriah et al., (2008). It is constructed to assign higher weights to mismatches on variables with more categories.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Eskin E., Arnold A., Prerau M., Portnoy L. and Stolfo S. (2002). A geometric framework for unsupervised anomaly detection. In D. Barbara and S. Jajodia (Eds): Applications of Data Mining in Computer Security, p. 78-100. Norwell: Kluwer Academic Publishers.

See Also

good1, good2, good3, good4, iof, lin, lin1, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.eskin <- eskin(data20)
```

`eval.plot` *Visualization of Evaluation Criteria*

Description

The function `eval.plot()` visualizes the values of seven evaluation criteria for the range of cluster solutions defined by the user in the **nomclust**, **evalclust** or **nomprox** functions. It also indicates the optimal number of clusters determined by these criteria. The charts for the seven evaluation criteria in the **nomclust** package.

Usage

```
eval.plot(
  x,
  criteria = "all",
  style = "greys",
  opt.col = "red",
  main = "Cluster Evaluation",
  ...
)
```

Arguments

`x`	An output of the `nomclust()` or `nomprox()` functions containing the `eval` and `opt` components.
`criteria`	A *character* string or character *vector* specifying the criteria that are going to be visualized. It can be selected one particular criterion, a vector of criteria or all the available criteria by typing `"all"`.
`style`	A *character* string or a *vector* of colors defines a graphical style of the produced plots. There are two predefined styles in the **nomclust** package, namely `"greys"` and `"dark"`, but a custom color scheme can be set by a user as a vector of a length four.
`opt.col`	An argument specifying a color that is used for the optimal number of clusters identification.
`main`	A *character* string with the chart title.
`...`	Other graphical arguments compatible with the generic `plot()` function.

Details

The function can be applied to the output of the `nomclust()`, `evalclust()` or `nomprox()` object containing a `eval` and `opt` components.

Value

The function returns a series of up to seven plots with evaluation criteria values and the graphical indication of the optimal numbers of clusters (for AIC, BIC, BK, PSFE, PSFM).

Author(s)

Jana Cibulkova and Zdenek Sulc.
Contact: <jana.cibulkova@vse.cz>

See Also

dend.plot, nomclust, evalclust, nomprox.

Examples

```
# sample data
data(data20)

# creating an object with results of hierarchical clustering
hca.object <- nomclust(data20, measure = "iof", eval = TRUE)

# a default series of plots
eval.plot(hca.object)

# changing the color indicating the optimum number of clusters
eval.plot(hca.object, opt.col= "darkorange")

# selecting only AIC and BIC criteria with the dark style
eval.plot(hca.object, criteria = c("AIC", "BIC"), style = "dark")
```

---

evalclust *Evaluation of Hierarchical Clustering for Nominal Data*

Description

The **evalclust** function calculates a set of evaluation criteria, see (Sulc et al., 2018) and provides the optimal number of clusters based on these criteria. It is primarily focused on the evaluation of hierarchical clustering results obtained by similarity measures different from the ones that occur in the **nomclust** package. Thus, it can serve for comparison of various similarity measures for categorical data.

Usage

```
evalclust(data, clusters)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.
clusters	A *data.frame* or a *list* of cluster memberships in a form of a sequence from the two-cluster solution to the maximal-cluster solution.

Value

The function returns a *list* with two components.

The `eval` component contains seven evaluation criteria in as vectors in a *list*. Namely, Within-cluster mutability coefficient (WCM), Within-cluster entropy coefficient (WCE), Pseudo F Indices based on the mutability (PSFM) and the entropy (PSFE), Bayessian (BIC) and Akaike (AIC) information criteria for categorical data and the BK index. To see them all in once, the form of a *data.frame* is more appropriate.

The `opt` component is present in the output together with the `eval` component. It displays the optimal number of clusters for the evaluation criteria from the `eval` component, except for WCM and WCE, where the optimal number of clusters is based on the elbow method.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Sulc Z., Cibulkova J., Prochazka J., Rezankova H. (2018). Internal Evaluation Criteria for Categorical Data in Hierarchical Clustering: Optimal Number of Clusters Determination, Metodoloski Zveski, 15(2), p. 1-20.

See Also

`nomclust`, `nomprox`, `eval.plot`.

Examples

```
# sample data
data(data20)

# creating an object with results of hierarchical clustering
hca.object <- nomclust(data20, measure = "iof", method = "average", clu.high = 7)

# the cluster memberships
data20.clu <- hca.object$mem

# obtaining evaluation criteria for the provided dataset and cluster memberships
data20.eval <- evalclust(data20, clusters = data20.clu)
```

good1 *Goodall 1 (G1) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the G1 similarity measure.

Usage

```
good1(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Goodall 1 similarity measure was presented in (Boriah et al., 2008). It is a simple modification of the original Goodall measure (Goodall, 1966). The measure assigns higher weights to infrequent matches.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Goodall V.D. (1966). A new similarity index based on probability. Biometrics, 22(4), p. 882.

See Also

eskin, good2, good3, good4, iof, lin, lin1, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.good1 <- good1(data20)
```

---

good2 *Goodall 2 (G2) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the G2 similarity measure.

Usage

```
good2(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Goodall 2 similarity measure was presented in (Boriah et al., 2008). It is a simple modification of the original Goodall measure (Goodall, 1966). The measure assigns weight to infrequent matches under the condition that there are also other categories, which are even less frequent than the examined one.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Goodall V.D. (1966). A new similarity index based on probability. Biometrics, 22(4), p. 882.

See Also

eskin, good1, good3, good4, iof, lin, lin1, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.good2 <- good2(data20)
```

---

good3 *Goodall 3 (G3) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the G3 similarity measure.

Usage

```
good3(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Goodall 3 similarity measure was presented in (Boriah et al., 2008). It is a simple modification of the original Goodall measure (Goodall, 1966). The measure assigns higher weight if the infrequent categories match regardless on frequencies of other categories.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Goodall V.D. (1966). A new similarity index based on probability. Biometrics, 22(4), p. 882.

See Also

eskin, good1, good2, good4, iof, lin, lin1, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.good3 <- good3(data20)
```

---

good4 *Goodall 4 (G4) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the G4 similarity measure.

Usage

```
good4(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Goodall 4 similarity measure was presented in (Boriah et al., 2008). It is a simple modification of the original Goodall measure (Goodall, 1966). It assigns higher weights to the frequent categories matches.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Goodall V.D. (1966). A new similarity index based on probability. Biometrics, 22(4), p. 882.

See Also

eskin, good1, good2, good3, iof, lin, lin1, morlini of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.good4 <- good4(data20)
```

---

iof *Inverse Occurence Frequency (IOF) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the IOF similarity measure.

Usage

```
iof(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The IOF (Inverse Occurrence Frequency) measure was originally constructed for the text mining tasks, see (Sparck-Jones, 1972), later, it was adjusted for categorical variables, see (Boriah et al., 2008). The measure assigns higher weight to mismatches on less frequent values and vice versa.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Spark-Jones K. (1972). A statistical interpretation of term specificity and its application in retrieval. In Journal of Documentation, 28(1), 11-21. Later: Journal of Documentation, 60(5) (2002), 493-502.

See Also

eskin, good1, good2, good3, good4, lin, lin1, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.iof <- iof(data20)
```

---

lin *Lin (LIN) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the LIN similarity measure.

Usage

```
lin(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Lin measure was introduced by Lin (1998) and presented in (Boriah et al., 2008). The measure assigns higher weights to more frequent categories in case of matches and lower weights to less frequent categories in case of mismatches.

Value

The function returns a dissimilarity matrix of the size `n` x `n`, where `n` is the number of objects in the original dataset in the argument `data`.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Lin D. (1998). An information-theoretic definition of similarity. In: ICML '98: Proceedings of the 15th International Conference on Machine Learning. San Francisco, p. 296-304.

See Also

`eskin`, `good1`, `good2`, `good3`, `good4`, `iof`, `lin1`, `morlini`, `of`, `sm`, `ve`, `vm`.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.lin <- lin(data20)
```

---

lin1 *Lin 1 (LIN1) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the LIN1 similarity measure.

Usage

```
lin1(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Lin 1 similarity measure was introduced in (Boriah et al., 2008) as a modification of the original Lin measure (Lin, 1998). In has a complex system of weights. In case of mismatch, lower similarity is assigned if either the mismatching values are very frequent or their relative frequency is in between the relative frequencies of mismatching values. Higher similarity is assigned if the mismatched categories are infrequent and there are a few other infrequent categories. In case of match, lower similarity is given for matches on frequent categories or matches on categories that have many other values of the same frequency. Higher similarity is given to matches on infrequent categories.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Lin D. (1998). An information-theoretic definition of similarity. In: ICML '98: Proceedings of the 15th International Conference on Machine Learning. San Francisco, p. 296-304.

See Also

eskin, good1, good2, good3, good4, iof, lin, morlini, of, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.lin1 <- lin1(data20)
```

`morlini` *Morlini and Zani's (MZ) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the MZ similarity measure.

Usage

```
morlini(data)
```

Arguments

`data`	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The MZ measure was originally introduced by Morlini and Zani (2012) under the name S2. The S2 measure was proposed. It is based on a binary-transformed dataset, so the **morlini** function must first create dummy-coded variables. The measure uses relative frequencies of categories of binary-coded variables, and it assigns higher weights to infrequent categories.

Value

The function returns a dissimilarity matrix of the size `n` x `n`, where `n` is the number of objects in the original dataset in the argument `data`.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Morlini I., Zani S. (2012). A new class of weighted similarity indices using polytomous variables. Journal of Classification, 29(2), p. 199-226.

See Also

`eskin`, `good1`, `good2`, `good3`, `good4`, `iof`, `lin`, `lin1`, `of`, `sm`, `ve`, `vm`.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.morlini <- morlini(data20)
```

---

nomclust *Hierarchical Cluster Analysis for Nominal Data*

Description

The nomclust() function runs hierarchical cluster analysis (HCA) with objects characterized by nominal (categorical) variables. It completely covers the clustering process, from the proximity matrix calculation to the evaluation of the quality of clustering. The function contains thirteen similarity measures for nominal data summarized in (Boriah et al., 2008) or introduced by Morlini and Zani in (Morlini and Zani, 2012), and by (Sulc and Rezankova, 2019). It offers three linkage methods that can be used for categorical data. The obtained clusters can be evaluated by seven evaluation criteria, see (Sulc et al., 2018). The output of the nomclust() function may serve as an input for visualization functions in the **nomclust** package.

Usage

```
nomclust(
  data,
  measure = "lin",
  method = "average",
  clu.high = 6,
  eval = TRUE,
  prox = 100,
  opt = TRUE
)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.
measure	A *character* string defining the similarity measure used for computation of proximity matrix in HCA: "eskin", "good1", "good2", "good3", "good4", "iof", "lin", "lin1", "morlini", "of", "sm", "ve", "vm".
method	A *character* string defining the clustering method. The following methods can be used: "average", "complete", "single".
clu.high	A *numeric* value expressing the maximal number of cluster for which the cluster memberships variables are produced.
eval	A *logical* operator; if TRUE, evaluation of the clustering results is performed.

`prox` A *logical* operator or a numeric value. If a logical value TRUE indicates that the proximity matrix is a part of the output. A numeric value (integer) of this argument indicates the maximal number of cases in a dataset for which a proximity matrix will occur in the output.

`opt` A *logical* operator; if TRUE, the time optimization method is run to substantially decrease computation time of the dissimilarity matrix calcation. Time optimalization method cannot be run if the proximity matrix is to be produced. In such a case, this parameter is automatically set to FALSE.

Value

The function returns a *list* with up to five components.

The `mem` component contains cluster membership partitions for the selected numbers of clusters in the form of a *list*.

The `eval` component contains seven evaluation criteria in as vectors in a *list*. Namely, Within-cluster mutability coefficient (WCM), Within-cluster entropy coefficient (WCE), Pseudo F Indices based on the mutability (PSFM) and the entropy (PSFE), Bayessian (BIC) and Akaike (AIC) information criteria for categorical data and the BK index. To see them all in once, the form of a *data.frame* is more appropriate.

The `opt` component is present in the output together with the `eval` component. It displays the optimal number of clusters for the evaluation criteria from the `eval` component, except for WCM and WCE, where the optimal number of clusters is based on the elbow method.

The `prox` component contains the dissimilarity matrix in a form of a *matrix*.

The `dend` component can be found in the output only together with the `prox` component. It contains all the necessary information for dendrogram creation.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V. and Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Morlini I. and Zani S. (2012). A new class of weighted similarity indices using polytomous variables. Journal of Classification, 29(2), p. 199-226.

Sulc Z., Cibulkova J., Prochazka J., Rezankova H. (2018). Internal Evaluation Criteria for Categorical Data in Hierarchical Clustering: Optimal Number of Clusters Determination, Metodoloski Zveski, 15(2), p. 1-20.

Sulc Z. and Rezankova H. (2019). Comparison of Similarity Measures for Categorical Data in Hierarchical Clustering. Journal of Classification. 2019, 35(1), p. 58-72. DOI: 10.1007/s00357-019-09317-5.

See Also

evalclust, nomprox, eval.plot, dend.plot.

Examples

```
# sample data
data(data20)

# creating an object with results of hierarchical clustering of
hca.object <- nomclust(data20, measure = "lin", method = "average",
 clu.high = 5, prox = TRUE, opt = FALSE)

# obtaining values of evaluation indices
data20.eval <- hca.object$eval

# getting the optimal numbers of clusters
data20.opt <- hca.object$opt

# extracting cluster membership variables
data20.mem <- hca.object$mem

# extracting cluster membership variables as a data frame
data20.mem <- as.data.frame(hca.object$mem)

# obtaining a proximity matrix
data20.prox <- hca.object$prox

# setting the maximal number of objects for which a proximity matrix is provided in the output to 30
hca.object <- nomclust(data20, measure = "lin", method = "average",
 clu.high = 5, prox = 30, opt = FALSE)

# generating of a larger dataset containing repeatedly occuring objects
set.seed(150)
sample150 <- sample(1:nrow(data20), 150, replace = TRUE)
data150 <- data20[sample150, ]

# running hierarchical clustering WITH the time optimization
start <- Sys.time()
hca.object.opt.T <- nomclust(data150, measure = "lin", opt = TRUE)
end <- Sys.time()
end - start

# running hierarchical clustering WITHOUT the time optimization
start <- Sys.time()
hca.object.opt.F <- nomclust(data150, measure = "lin", opt = FALSE)
end <- Sys.time()
end - start
```

nomprox *Hierarchical Cluster Analysis for Nominal Data Based on a Proximity Matrix*

Description

The nomprox() function performs hierarchical cluster analysis in situations when the proximity (dissimilarity) matrix was calculated externally. For instance, in a different R package, in an own-created function, or in other software. It offers three linkage methods that can be used for categorical data. The obtained clusters can be evaluated by seven evaluation indices, see (Sulc et al., 2018).

Usage

```
nomprox(diss, data = NULL, method = "average", clu.high = 6, eval = TRUE)
```

Arguments

diss	A proximity matrix or a dist object calculated from the dataset defined in a parameter data.
data	A *data.frame* or a *matrix* with cases in rows and variables in colums.
method	A *character* string defining the clustering method. The following methods can be used: "average", "complete", "single".
clu.high	A *numeric* value expressing the maximal number of cluster for which the cluster memberships variables are produced.
eval	A *logical* operator; if TRUE, evaluation of clustering results is performed.

Value

The function returns a list with up to three components:

The mem component contains cluster membership partitions for the selected numbers of clusters in the form of a *list*.

The eval component contains seven evaluation criteria in as vectors in a *list*. Namely, Within-cluster mutability coefficient (WCM), Within-cluster entropy coefficient (WCE), Pseudo F Indices based on the mutability (PSFM) and the entropy (PSFE), Bayessian (BIC) and Akaike (AIC) information criteria for categorical data and the BK index. To see them all in once, the form of a *data.frame* is more appropriate.

The opt component is present in the output together with the eval component. It displays the optimal number of clusters for the evaluation criteria from the eval component, except for WCM and WCE, where the optimal number of clusters is based on the elbow method.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Sulc Z., Cibulkova J., Prochazka J., Rezankova H. (2018). Internal Evaluation Criteria for Categorical Data in Hierarchical Clustering: Optimal Number of Clusters Determination, Metodoloski Zveski, 15(2), p. 1-20.

See Also

nomclust, evalclust, eval.plot.

Examples

```
# sample data
data(data20)

# computation of a dissimilarity matrix using the iof similarity measure
diss.matrix <- iof(data20)

# creating an object with results of hierarchical clustering
hca.object <- nomprox(diss = diss.matrix, data = data20, method = "complete",
 clu.high = 5, eval = TRUE)
```

---

of *Occurence Frequency (OF) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the OF similarity measure.

Usage

```
of(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The OF (Occurrence Frequency) measure was originally constructed for the text mining tasks, see (Sparck-Jones, 1972), later, it was adjusted for categorical variables, see (Boriah et al., 2008) It assigns higher weight to mismatches on less frequent values and otherwise.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Spark-Jones K. (1972). A statistical interpretation of term specificity and its application in retrieval. In Journal of Documentation, 28(1), p. 11-21. Later: Journal of Documentation, 60(5) (2002), p. 493-502.

See Also

eskin, good1, good2, good3, good4, iof, lin, lin1, morlini, sm, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.of <- of(data20)
```

`sm` *Simple Matching Coefficient (SM)*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the SM similarity measure.

Usage

```
sm(data)
```

Arguments

`data`	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The simple matching coefficient (Sokal, 1958) represents the simplest way of measuring similarity. It does not impose any weights. By a given variable, it assigns the value 1 in case of match and value 0 otherwise.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Sokal R., Michener C. (1958). A statistical method for evaluating systematic relationships. In: Science bulletin, 38(22), The University of Kansas.

See Also

eskin, good1, good2, good3, good4, iof, lin, lin1, morlini, of, ve, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.sm <- sm(data20)
```

---

ve *Variable Entropy (VE) Measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the VE similarity measure.

Usage

```
ve(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Variable Entropy similarity measure was introduced in (Sulc and Rezankova, 2019). It treats the similarity between two categories based on the within-cluster variability expressed by the normalized entropy. The measure assigns higher weights to rare categories.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Boriah S., Chandola V., Kumar V. (2008). Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, p. 243-254.

Sulc Z. and Rezankova H. (2019). Comparison of Similarity Measures for Categorical Data in Hierarchical Clustering. Journal of Classification. 2019, 35(1), p. 58-72. DOI: 10.1007/s00357-019-09317-5.

See Also

eskin, good1, good2, good3, good4, iof, lin, lin1, morlini, of, sm, vm.

Examples

```
# sample data
data(data20)

# dissimilarity matrix calculation
prox.ve <- ve(data20)
```

---

vm *Variable Mutability (VM) measure*

Description

A function for calculation of a proximity (dissimilarity) matrix based on the VM similarity measure.

Usage

```
vm(data)
```

Arguments

data	A *data.frame* or a *matrix* with cases in rows and variables in colums.

Details

The Variable Mutability similarity measure was introduced in (Sulc and Rezankova, 2019). It treats the similarity between two categories based on the within-cluster variability expressed by the normalized mutability. The measure assigns higher weights to rarer categories.

Value

The function returns a dissimilarity matrix of the size n x n, where n is the number of objects in the original dataset in the argument data.

Author(s)

Zdenek Sulc.
Contact: <zdenek.sulc@vse.cz>

References

Sulc Z. and Rezankova H. (2019). Comparison of Similarity Measures for Categorical Data in Hierarchical Clustering. Journal of Classification. 2019, 35(1), p. 58-72. DOI: 10.1007/s00357-019-09317-5.

See Also

eskin, good1, good2, good3, good4, iof, lin, lin1, morlini, of, sm, ve.

Examples

```
#sample data
data(data20)

# dissimilarity matrix calculation
prox.vm <- vm(data20)
```

Index

∗ **clustering**
CA.methods, 2
∗ **datasets**
data20, 2

CA.methods, 2

data20, 2
dend.plot, 3, *7*, *20*

eskin, 4, *9*, *11–17*, *23–26*
eval.plot, *4*, 6, *8*, *20*, *22*
evalclust, *7*, 7, *20*, *22*

good1, *5*, 9, *11–17*, *23–26*
good2, *5*, *9*, 10, *12–17*, *23–26*
good3, *5*, *9*, *11*, 11, *13–17*, *23–26*
good4, *5*, *9*, *11, 12*, 12, *14–17*, *23–26*

iof, *5*, *9*, *11–13*, 13, *15–17*, *23–26*

lin, *5*, *9*, *11–14*, 14, *16, 17*, *23–26*
lin1, *5*, *9*, *11–15*, 15, *17*, *23–26*

morlini, *5*, *9*, *11–16*, 17, *23–26*

nomclust, *4*, *7, 8*, 18, *22*
nomprox, *4*, *7, 8*, *20*, 21

of, *5*, *9*, *11–17*, 22, *24–26*

sm, *5*, *9*, *11–17*, *23*, 23, *25, 26*

ve, *5*, *9*, *11–17*, *23, 24*, 24, *26*
vm, *5*, *9*, *11–17*, *23–25*, 25