Package 'CrossClustering'

July 30, 2018

Type Package

Title A Partial Clustering Algorithm

Version 4.0.3

Date 2018-07-29

Maintainer Paola Tellaroli <paola.tellaroli@gmail.com>

Description Provide the CrossClustering algorithm (Tellaroli et al. (2016) <doi:10.1371/journal.pone.0152333>), which is a partial clustering algorithm that combines the Ward's minimum variance and Complete Linkage algorithms, providing automatic estimation of a suitable number of clusters and identification of outlier elements.

License GPL (>= 3)

URL https://CRAN.R-project.org/package=CrossClustering

RoxygenNote 6.0.1

BugReports https://github.com/CorradoLanera/CrossClustering/issues

Depends R (>= 3.1)

Imports cluster, mclust, flip, magrittr, purrr, utils, assertive, crayon, glue, cli, dplyr

Suggests testthat, covr, roxygen2

Encoding UTF-8

LazyData true

NeedsCompilation no

Author Paola Tellaroli [cre, aut],
Marco Bazzi [aut],
Michele Donato [aut],
Livio Finos [aut],
Philippe Courcoux [aut],
Corrado Lanera [aut]

Repository CRAN

Date/Publication 2018-07-30 15:50:06 UTC

R topics documented:

`ari` *Computes the adjusted Rand index and the confidence interval, comparing two classifications from a contingency table.*

Description

Computes the adjusted Rand index and the confidence interval, comparing two classifications from a contingency table.

print method for ari class

Usage

```
ari(mat, alpha = 0.05, digits = 2)

## S3 method for class 'ari'
print(x, ...)
```

Arguments

`mat`	A matrix of integers representing the contingency table of reference
`alpha`	A single number strictly included between 0 and 1 representing the significance level of interest. (default is 0.05)
`digits`	An integer for the returned significant digits to return (default is 2)
`x`	an object used to select a method.
`...`	further arguments passed to or from other methods.

Details

The adjusted Rand Index (ARI) should be interpreted as follows:

ARI >= 0.90 excellent recovery; 0.80 =< ARI < 0.90 good recovery; 0.65 =< ARI < 0.80 moderate recovery; ARI < 0.65 poor recovery.

As the confidence interval is based on the approximation to the Normal distribution, it is recommended to trust in the confidence interval only in cases of total number of object clustered greater than 100.

Value

An object of class `ari` with the following elements:

`AdjustedRandIndex`
: The adjusted Rand Index

`CI`
: The confidence interval

Methods (by generic)

- `print`:

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>;

References

L. Hubert and P. Arabie (1985) Comparing partitions, Journal of Classification, 2, 193-218.

E.M. Qannari, P. Courcoux and Faye P. (2014) Significance test of the adjusted Rand index. Application to the free sorting task, Food Quality and Preference, (32)93-97

M.H. Samuh, F. Leisch, and L. Finos (2014), Tests for Random Agreement in Cluster Analysis, Statistica Applicata-Italian Journal of Applied Statistics, vol. 26, no. 3, pp. 219-234.

D. Steinley (2004) Properties of the Hubert-Arabie Adjusted Rand Index, Psychological Methods, 9(3), 386-396

D. Steinley, M.J. Brusco, L. Hubert (2016) The Variance of the Adjusted Rand Index, Psychological Methods, 21(2), 261-272

Examples

```
#### This example compares the adjusted Rand Index as computed on the
### partitions given by Ward's algorithm with the ground truth on the
### famous Iris data set by the adjustedRandIndex function
### {mclust package} and by the ari function.

library(CrossClustering)
library(mclust)

clusters <- iris[-5] %>%
```

```
  dist %>%
  hclust(method = 'ward.D') %>%
  cutree(k = 3)

ground_truth <- iris[[5]] %>% as.numeric()

mc_ari <- adjustedRandIndex(clusters, ground_truth)
mc_ari

ari_cc <- table(ground_truth, clusters) %>%
  ari(digits = 7)
ari_cc

all.equal(mc_ari, unclass(ari_cc), check.attributes = FALSE)
```

`cc_crossclustering` *A partial clustering algorithm with automatic estimation of the number of clusters and identification of outliers*

Description

This function performs the CrossClustering algorithm. This method combines the Ward's minimum variance and Complete-linkage (default, useful for finding spherical clusters) or Single-linkage (useful for finding elongated clusters) algorithms, providing automatic estimation of a suitable number of clusters and identification of outlier elements.

print method for crossclustering class

Usage

```
cc_crossclustering(dist, k_w_min = 2, k_w_max = attr(dist, "Size") - 2,
  k2_max = k_w_max + 1, out = TRUE, method = c("complete", "single"))

## S3 method for class 'crossclustering'
print(x, ...)
```

Arguments

`dist`	A dissimilarity structure as produced by the function `dist`
`k_w_min`	[int] Minimum number of clusters for the Ward's minimum variance method. By default is set equal 2
`k_w_max`	[int] Maximum number of clusters for the Ward's minimum variance method (see details)
`k2_max`	[int] Maximum number of clusters for the Complete/Single-linkage method. It can not be equal or greater than the number of elements to cluster (see details)
`out`	[lgl] If `TRUE` (default) outliers must be searched (see details)

method	[chr] "complete" (default) or "single". CrossClustering combines Ward's algorithm with Complete-linkage if method is set to "complete", otherwise (if method is set to 'single') Single-linkage will be used.
x	an object used to select a method.
...	further arguments passed to or from other methods.

Details

See cited document for more details.

Value

A list of objects describing characteristics of the partitioning as follows:

Optimal_cluster	number of clusters
Cluster_list	a list of clusters; each element of this lists contains the indices of the elements belonging to the cluster
Silhouette	the average silhouette width over all the clusters
n_total	total number of input elements
n_clustered	number of input elements that have actually been clustered

Methods (by generic)

- print:

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>;; Marco Bazzi, <bazzi [at] stat [dot] unipd [dot] it>; Michele Donato, <mdonato [at] stanford [dot] edu>

References

Tellaroli P, Bazzi M., Donato M., Brazzale A. R., Draghici S. (2016). Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters. PLoS ONE 11(3): e0152333. doi:10.1371/journal.pone.0152333

#' Tellaroli P, Bazzi M., Donato M., Brazzale A. R., Draghici S. (2017). E1829: Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters. CMStatistics 2017, London 16-18 December, Book of Abstracts (ISBN 978-9963-2227-4-2)

Examples

```
library(CrossClustering)

#### Example of Cross-Clustering as in reference paper
#### method = "complete"

data(toy)
```

```
### toy is transposed as we want to cluster samples (columns of the
### original matrix)
toy_dist <- t(toy) %>%
  dist(method = "euclidean")

### Run CrossClustering
cc_crossclustering(toy_dist,
  k_w_min = 2,
  k_w_max = 5,
  k2_max  = 6,
  out     = TRUE
)

#### Simulated data as in reference paper
#### method = "complete"
set.seed(10)
sg <- c(500, 250, 700, 300, 100)

# 5 clusters

t <- matrix(0, nrow = 5, ncol = 5)
t[1, ] <- rep(6, 5)
t[2, ] <- c( 0,  5, 12, 13, 15)
t[3, ] <- c(15, 11,  9,  5,  0)
t[4, ] <- c( 6, 12, 15, 10,  5)
t[5, ] <- c(12, 17,  3,  7, 10)

t_mat <- NULL
for (i in seq_len(nrow(t))){
  t_mat <- rbind(
    t_mat,
    matrix(rep(t[i, ], sg[i]), nrow = sg[i], byrow = TRUE)
  )
}

data_15 <- matrix(NA, nrow = 2000, ncol = 5)
data_15[1:1850, ] <- matrix(abs(rnorm(sum(sg) * 5, sd = 1.5)),
  nrow = sum(sg),
  ncol = 5
) + t_mat

set.seed(100) # simulate outliers
data_15[1851:2000, ] <- matrix(
  runif(n = 150 * 5, min = 0, max = max(data_15, na.rm = TRUE)),
  nrow = 150,
  ncol = 5
)

### Run CrossClustering
cc_crossclustering(dist(data_15),
  k_w_min = 2,
  k_w_max = 19,
  k2_max  = 20,
```

```
  out     = TRUE
)


#### Correlation-based distance is often used in gene expression time-series
### data analysis. Here there is an example, using the "complete" method.

data(nb_data)
nb_dist <- as.dist(1 - abs(cor(t(nb_data))))
cc_crossclustering(dist = nb_dist, k_w_max = 20, k2_max = 19)




#### method = "single"
### Example on a famous shape data set
### Two moons data

data(twomoons)

moons_dist <- twomoons[, 1:2] %>%
  dist(method = "euclidean")

cc_moons <- cc_crossclustering(moons_dist,
  k_w_max = 9,
  k2_max  = 10,
  method  = 'single'
)

moons_col <- cc_get_cluster(cc_moons)
plot(twomoons[, 1:2], col = moons_col,
  pch      = 19,
  xlab     = "",
  ylab     = "",
  main     = "CrossClustering-Single"
)

### Worms data
data(worms)

worms_dist <- worms[, 1:2] %>%
  dist(method = "euclidean")

cc_worms <- cc_crossclustering(worms_dist,
  k_w_max = 9,
  k2_max  = 10,
  method  = 'single'
)

worms_col <-  cc_get_cluster(cc_worms)

plot(worms[, 1:2], col = worms_col,
  pch      = 19,
```

```
  xlab     = "",
  ylab     = "",
  main     = "CrossClustering-Single"
)


### CrossClustering-Single is not affected to chain-effect problem

data(chain_effect)

chain_dist <- chain_effect %>%
  dist(method = "euclidean")
cc_chain <- cc_crossclustering(chain_dist,
  k_w_max = 9,
  k2_max  = 10,
  method  = 'single'
)

chain_col <- cc_get_cluster(cc_chain)

plot(chain_effect, col = chain_col,
  pch  = 19,
  xlab = "",
  ylab = "",
  main = "CrossClustering-Single"
)
```

`cc_get_cluster` *Provides the vector of clusters' ID to which each element belong to.*

Description

Provides the vector of clusters' ID to which each element belong to.

Usage

```
cc_get_cluster(x, n_elem)

## Default S3 method:
cc_get_cluster(x, n_elem)

## S3 method for class 'crossclustering'
cc_get_cluster(x, n_elem)
```

Arguments

`x`	list of clustered elements or a `crossclustering` object
`n_elem`	total number of elements clustered (ignored if x is of class `crossclustering`)

Value

An integer vector of clusters to which the elements belong (‘1‘ for the outliers, ID + 1 for the others).

Methods (by class)

- `default`: default method for `cc_get_cluster`.
- `crossclustering`: automatically extract inputs from a `crossclustering` object

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>;; Marco Bazzi, <bazzi [at] stat [dot] unipd [dot] it>; Michele Donato, <mdonato [at] stanford [dot] edu>.

References

Tellaroli P, Bazzi M., Donato M., Brazzale A. R., Draghici S. (2016). Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters. PLoS ONE 11(3): e0152333. doi:10.1371/journal.pone.0152333

Examples

```
library(CrossClustering)

data(toy)

### toy is transposed as we want to cluster samples (columns of the
### original matrix)
toy_dist <- t(toy) %>% dist(method = "euclidean")

### Run CrossClustering
toyres <- cc_crossclustering(toy_dist,
  k_w_min = 2,
  k_w_max = 5,
  k2_max  = 6,
  out     = TRUE
)

### cc_get_cluster
cc_get_cluster(toyres[], 7)


### cc_get_cluster directly from a crossclustering object
cc_get_cluster(toyres)
```

cc_test_ari *A test for testing the null hypothesis of random agreement (i.e., adjusted Rand Index equal to 0) between two partitions.*

Description

A test for testing the null hypothesis of random agreement (i.e., adjusted Rand Index equal to 0) between two partitions.

Usage

```
cc_test_ari(ground_truth, partition)
```

Arguments

ground_truth	[int] A vector of the actual membership of elements in clusters
partition	The partition coming from a clustering algorithm

Value

A list with six elements:

Rand	the Rand Index
ExpectedRand	expected value of Rand Index
AdjustedRand	Adjusted Rand Index
varARI	variance of Rand Index
NARI	NARI
p-value	the p-value of the test

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>; Philippe Courcoux, <philippe [dot] courcoux [at] oniris-nantes [dot] fr>

References

E_M. Qannari, P. Courcoux and Faye P. (2014) Significance test of the adjusted Rand index. Application to the free sorting task, Food Quality and Preference, (32)93-97

L. Hubert and P. Arabie (1985) Comparing partitions, Journal of Classification, 2, 193-218.

Examples

```
library(CrossClustering)

clusters <- iris[-5] %>%
  dist %>%
  hclust(method = 'ward.D') %>%
  cutree(k = 3)

ground_truth <- iris[[5]] %>% as.numeric()

CrossClustering:::cc_test_ari(ground_truth, clusters)
```

`cc_test_ari_permutation` *A permutation test for testing the null hypothesis of random agreement (i.e., adjusted Rand Index equal to 0) between two partitions.*

Description

A permutation test for testing the null hypothesis of random agreement (i.e., adjusted Rand Index equal to 0) between two partitions.

Usage

```
cc_test_ari_permutation(ground_truth, partition)
```

Arguments

`ground_truth`	[int] A vector of the actual membership of elements in clusters
`partition`	The partition coming from a clustering algorithm

Value

A data_frame with two columns:

`ari`	the adjusted Rand Index
`p_value`	the p-value of the test

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>; Livio Finos, <livio [dot] finos [at] unipd [dot] it>

References

Samuh M. H., Leisch F., and Finos L. (2014), Tests for Random Agreement in Cluster Analysis, Statistica Applicata-Italian Journal of Applied Statistics, vol. 26, no. 3, pp. 219-234.

L. Hubert and P. Arabie (1985) Comparing partitions, Journal of Classification, 2, 193-218.

Examples

```
library(CrossClustering)

clusters <- iris[-5] %>%
  dist %>%
  hclust(method = 'ward.D') %>%
  cutree(k = 3)

ground_truth <- iris[[5]] %>% as.numeric()

CrossClustering:::cc_test_ari_permutation(ground_truth, clusters)
```

---

chain_effect *A toy dataset for illustrating the chain effect.*

Description

A toy dataset for illustrating the chain effect.

Usage

```
chain_effect
```

Format

A data frame with 28 rows and 2 variables:

`X` numx coordinates 0 is negative.

`Y` numy coordinates.

---

consensus_cluster *Get clusters which reach max consensus*

Description

Computes the consensus between Ward's minimum variance and Complete-linkage (or Single-linkage) algorithms (i.e., the number of elements classified together by both algorithms).

Usage

```
consensus_cluster(k, cluster_ward, cluster_other)
```

Arguments

k	[int] a vector containing the number of clusters for Ward and for Complete-linkage (or Single-linkage) algorithms, respectively
cluster_ward	an object of class hclust for the Ward algorithm
cluster_other	an object of class hclust for the Complete-linkage (or Single-linkage) algorithm

Value

an object of class consensus_cluster with the following elements:

elements	list of the elements belonging to each cluster
;	
A_star	contingency table of the clustering
;	
max_consensus	maximum clustering consensus
.	

Author(s)

Paola Tellaroli, <paola [dot] tellaroli [at] unipd [dot] it>;; Marco Bazzi, <bazzi [at] stat [dot] unipd [dot] it>; Michele Donato, <mdonato [at] stanford [dot] edu>.

References

Tellaroli P, Bazzi M., Donato M., Brazzale A. R., Draghici S. (2016). Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters. PLoS ONE 11(3): e0152333. doi:10.1371/journal.pone.0152333

Examples

```
library(CrossClustering)

data(toy)

### toy is transposed as we want to cluster samples (columns of the
### original matrix)
toy_dist <- t(toy) %>% dist(method = "euclidean")

### Hierarchical clustering
cluster_ward    <- toy_dist %>% hclust(method = "ward.D")
cluster_other <- toy_dist %>% hclust(method = "complete")


### consensus_cluster
CrossClustering:::consensus_cluster(c(3, 4),
  cluster_ward,
  cluster_other
```

```
)
```

`is_zero` *Check for zero*

Description

Check if a given, single, number is 0 or not

Usage

```
is_zero(num)
```

Arguments

`num`	a numerical vector of length one

Value

a boolean, TRUE if num is 0

Examples

```
CrossClustering:::is_zero(1)
CrossClustering:::is_zero(0)
```

`nb_data` *RNA-Seq dataset example*

Description

'nb_data' contains a subset of a bigger normalized negative binomial simulated dataset.

Usage

```
nb_data
```

Format

A data frame with 100 observations on 36 numeric variables.

Details

This dataset is part of a larger simulated and normalized dataset with 2 experimental groups, 6 time-points and 3 replicates. Simulation has been done by using a negative binomial distribution. The first 20 genes are simulated with changes among time.

Source

Data included in the bioconductor package 'maSigPro'. https://doi.org/doi:10.18129/B9.bioc.maSigPro

---

`prune_zero_tail` *Prune tail made of zeros*

Description

Given a diagonal matrix which is supposed to have no non-zero entry in the diagonal after the first one (if any) the function returns the diagonal (sub-)matrix without the columns and the row corresponding to the zero-entries in the diagonal (if any).

Usage

```
prune_zero_tail(diag_mat)
```

Arguments

`diag_mat`	a diagonal matrix which must satisfy the following property: in the diagonal, every element after a zero is a zero.

Value

a diagonal matrix without zeros in the diagonal, composed by the first rows and columns of the original matrix with non zeros in the diagonal (which are also the only ones)

Examples

```
diag_mat <- diag(c(1, 2, 3, 0, 0, 0, 0))
prune_zero_tail(diag_mat)
```

`reverse_table` *Reverse the process of create a contingency table*

Description

Reverse the process of create a contingency table

Usage

```
reverse_table(x)
```

Arguments

`x`	a contingency table

Value

a list of 2 vector corresponding to the unrolled table

Examples

```
clust_1 <- iris[, 1:4] %>% dist() %>% hclust() %>% cutree(k = 3)
clust_2 <- iris[, 1:4] %>% dist() %>% hclust() %>% cutree(k = 4)
cont_table <- table(clust_1, clust_2)

reverse_table(cont_table)
```

---

toy *A toy example matrix*

Description

A toy example matrix

Usage

```
toy
```

Format

A matrix of 10 row and 7 columns

---

twomoons *A famous shape data set containing two clusters with two moons shapes and outliers*

Description

A famous shape data set containing two clusters with two moons shapes and outliers

Usage

```
twomoons
```

Format

A data frame with 52 rows and 3 variables:

`x` numx coordinates

`y` numy coordinates.

`clusters` integercluster membership (outliers classified as 3rd cluster).

worms — *A famous shape data set containing two clusters with two worms shapes and outliers*

Description

A famous shape data set containing two clusters with two worms shapes and outliers

Usage

```
worms
```

Format

A data frame with 87 rows and 3 variables:

`x` numx coordinates

`y` numy coordinates.

`cluster` integercluster membership (outliers classified as 3rd cluster).

Index