Package: clevr (via r-universe)

September 7, 2024

Type Package

Title Clustering and Link Prediction Evaluation in R

Version 0.1.2

Date 2023-09-16

Maintainer Neil Marchant <ngmarchant@gmail.com>

Description Tools for evaluating link prediction and clustering algorithms with respect to ground truth. Includes efficient implementations of common performance measures such as pairwise precision/recall, cluster homogeneity/completeness, variation of information, Rand index etc.

License GPL-2

Encoding UTF-8

Depends R (>= 3.0.2)

Imports Rcpp (>= 1.0.5), stats, Matrix

LinkingTo Rcpp, BH (>= 1.69.0)

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

Suggests testthat

URL https://github.com/cleanzr/clevr

BugReports https://github.com/cleanzr/clevr/issues

Collate 'RcppExports.R' 'clevr.R' 'measures_clusterings.R' 'transformations.R' 'measures_pairs.R'

Repository https://cleanzr.r-universe.dev

RemoteUrl https://github.com/cleanzr/clevr

RemoteRef HEAD

RemoteSha 7757279d021aa91fc3d6efdcda90070f64d25960

Contents

`accuracy_pairs` *Accuracy of Linked Pairs*

Description

Computes the accuracy of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
accuracy_pairs(true_pairs, pred_pairs, num_pairs, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.
`num_pairs`	the total number of coreferent and non-coreferent pairs, excluding equivalent pairs with reversed ids.

`ordered` whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The accuracy is defined as:

$$\frac{|T \cap P| + |T' \cap P'|}{N}$$

where:

- T is the set of true coreferent pairs,
- P is the set of predicted coreferent pairs,
- T' is the set of true non-coreferent pairs,
- P' is the set of predicted non-coreferent pairs, and
- N is the total number of coreferent and non-coreferent pairs.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
accuracy_pairs(true_pairs, pred_pairs, num_pairs)
```

---

adj_rand_index *Adjusted Rand Index Between Clusterings*

Description

Computes the adjusted Rand index (ARI) between two clusterings, such as a predicted and ground truth clustering.

Usage

```
adj_rand_index(true, pred)
```

Arguments

`true` ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.

`pred` predicted clustering represented as a membership vector.

Details

The adjusted Rand index (ARI) is a variant of the Rand index (RI) which is corrected for chance using the Permutation Model for clusterings. It is related to the RI as follows:

$$\frac{RI - E(RI)}{1 - E(RI)},$$

where $E(RI)$ is the expected value of the RI under the Permutation Model. Unlike the RI, the ARI takes values in the range -1 to 1. A value of 1 indicates that the clusterings are identical, while a value of 0 indicates the clusterings are drawn randomly independent of one another.

References

Hubert, L., Arabie, P. "Comparing partitions." *Journal of Classification* **2**, 193–218 (1985). doi:10.1007/BF01908075

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
adj_rand_index(true, pred)
```

`balanced_accuracy_pairs`

Balanced Accuracy of Linked Pairs

Description

Computes the balanced accuracy of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
balanced_accuracy_pairs(true_pairs, pred_pairs, num_pairs, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.
`num_pairs`	the total number of coreferent and non-coreferent pairs, excluding equivalent pairs with reversed ids.
`ordered`	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The balanced accuracy is defined as:

$$\frac{\frac{|T \cap P|}{|P|} + \frac{|T' \cap P'|}{|P'|}}{2}$$

where:

- T is the set of true coreferent pairs,
- P is the set of predicted coreferent pairs,
- T' is the set of true non-coreferent pairs, and
- P' is the set of predicted non-coreferent pairs.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
balanced_accuracy_pairs(true_pairs, pred_pairs, num_pairs)
```

---

canonicalize_pairs *Canonicalize element pairs*

Description

Coerce a collection of element pairs into canonical form. Facilitates testing of equivalence.

Usage

```
canonicalize_pairs(pairs, ordered = FALSE)
```

Arguments

pairs	a matrix or data.frame of element pairs where rows correspond to element pairs and columns correspond to element identifiers.
ordered	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Value

Returns the element pairs in canonical form, so that:

- the first element id precedes the second element id lexicographically if `ordered = FALSE`—i.e. pair (3, 2) becomes pair (2, 3);
- pairs with missing element ids are removed;
- duplicate pairs are removed; and
- the rows in the matrix/data.frame pairs are sorted lexicographically by the first element id, then by the second element id.

Examples

```
messy_pairs <- rbind(c(2,1), c(1,2), c(3,1), c(1,2))
clean_pairs <- canonicalize_pairs(messy_pairs)
all(rbind(c(1,2), c(1,3)) == clean_pairs) # duplicates removed and order fixed
```

`clusters_to_membership` *Transform Clustering Representations*

Description

Transform between different representations of a clustering.

Usage

```
clusters_to_membership(clusters, elem_ids = NULL, clust_ids = NULL)

membership_to_clusters(membership, elem_ids = NULL, clust_ids = NULL)

clusters_to_pairs(clusters)

membership_to_pairs(membership, elem_ids = NULL)

pairs_to_membership(pairs, elem_ids)

pairs_to_clusters(pairs, elem_ids)
```

Arguments

`clusters`	a representation of a clustering as a list of vectors, where the i-th vector contains the identifiers of elements assigned to the i-th cluster. If `clust_ids` is specified (see below), the i-th cluster is identified according to the corresponding entry in `clust_ids`. Otherwise the i-th cluster is identified according it's name (if `clusters` is a named list) or its integer index i.
`elem_ids`	a vector specifying the complete set of identifiers for the cluster elements in canonical order. Optional for all functions excluding `pairs_to_membership` and `pairs_to_clusters`.
`clust_ids`	a vector specifying the complete set of identifiers for the clusters in canonical order. Optional for all functions.
`membership`	a representation of a clustering as a membership vector, where the i-th entry contains the cluster identifier for the i-th element. If `elem_ids` is specified (see below), the i-th element is identified according to the corresponding entry in `elem_ids`. Otherwise the i-th element is identified according it's name (if `members` is a named vector) or its integer index i.

pairs — a representation of a clustering as a matrix or data.frame containing all pairs of elements that are co-clustered. The rows index of the matrix/data.frame index pairs and columns index the identifiers of the constituent elements. The elem_ids argument (see below) must be specified in order to recover singleton clusters (containing a single element).

Value

clusters_to_membership and pairs_to_membership both return a membership vector representation of the clustering. The order of the elements is taken from elem_ids if specified, otherwise the elements are ordered lexicographically by their identifiers. For pairs_to_membership, the cluster identifiers cannot be recovered and are taken to be integers.

membership_to_clusters and pairs_to_clusters both return a representation of the clustering as a list of vectors. The order of the clusters is taken from clust_ids if specified, otherwise the clusters are ordered lexicographically by their identifiers. For pairs_to_clusters, the cluster identifiers cannot be recovered and are taken to be integers.

clusters_to_pairs and membership_to_pairs both return a representation of the clustering as a matrix of element pairs that are co-clustered. This representation results in loss of information, as singleton clusters (with one element) and cluster identifiers are not represented.

Examples

```
## A clustering of three items represented as a membership vector
m <- c("Item1" = 1, "Item2" = 2, "Item3" = 1)

# Transform to list of clusters
membership_to_clusters(m)
# Specify different identifiers for the items
membership_to_clusters(m, elem_ids = c(1, 2, 3))
# Transform to array of pairs that are co-clustered
membership_to_pairs(m)

## A clustering represented as a list of clusters
cl <- list("ClustA" = c(1,3), "ClustB" = c(2))

# Transform to membership vector representation
clusters_to_membership(cl)
# Transform to array of pairs that are co-clustered
clusters_to_pairs(cl)

## A clustering (incompletely) represented as an array of pairs that
## are co-clustered
p <- rbind(c(1,3)) # pairs of elements in the same cluster
ids <- c(1,2,3)    # necessary to specify set of all elements

# Transform to membership vector representation
pairs_to_membership(p, ids)
# Transform to list of clusters
pairs_to_clusters(p, ids)
```

`completeness` *Completeness Between Clusterings*

Description

Computes the completeness between two clusterings, such as a predicted and ground truth clustering.

Usage

```
completeness(true, pred)
```

Arguments

`true`	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
`pred`	predicted clustering represented as a membership vector.

Details

Completeness is an entropy-based measure of the similarity between two clusterings, say t and p. The completeness is high if *all* members of a given cluster in t are assigned to a single cluster in p. The completeness ranges between 0 and 1, where 1 indicates perfect completeness.

References

Rosenberg, A. and Hirschberg, J. "V-measure: A conditional entropy-based external cluster evaluation measure." *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), (2007).

See Also

`homogeneity` evaluates the *homogeneity*, which is a dual measure to *completeness*. `v_measure` evaluates the harmonic mean of *completeness* and *homogeneity*.

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
completeness(true, pred)
```

contingency_table_clusters

Contingency Table for Clusterings

Description

Compute the contingency table for a *predicted* clustering given a *ground truth* clustering.

Usage

```
contingency_table_clusters(true, pred)
```

Arguments

`true`	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
`pred`	predicted clustering represented as a membership vector.

Value

Returns a table C (stored as a sparse matrix) such that C_{ij} counts the number of elements assigned to cluster i in `pred` and cluster j in `true`.

See Also

`eval_report_clusters` computes common evaluation measures derived from the output of this function.

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
contingency_table_clusters(true, pred)
```

contingency_table_pairs

Binary Contingency Table for Linked Pairs

Description

Compute the binary contingency table for a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
contingency_table_pairs(
  true_pairs,
  pred_pairs,
  num_pairs = NULL,
  ordered = FALSE
)
```

Arguments

true_pairs	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
pred_pairs	set of predicted coreferent pairs, following the same specification as `true_pairs`.
num_pairs	the total number of coreferent and non-coreferent pairs, excluding equivalent pairs with reversed ids. If not provided, the true negative cell will be set to NA.
ordered	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Value

Returns a 2×2 contingency table of the form:

```
            Truth
Prediction   TRUE  FALSE
     TRUE      TP     FP
     FALSE     FN     TN
```

See Also

The membership_to_pairs and clusters_to_pairs functions can be used to transform other clustering representations into lists of pairs, as required by this function. The eval_report_pairs function computes common evaluation measures derived from binary contingency matrices, like the ones output by this function.

Examples

```
### Example where pairs/edges are undirected
# ground truth is 3-clique
true_pairs <- rbind(c(1,2), c(2,3), c(1,3))
# prediction misses one edge
pred_pairs <- rbind(c(1,2), c(2,3))
# total number of pairs assuming 3 elements
num_pairs <- 3 * (3 - 1) / 2
eval_report_pairs(true_pairs, pred_pairs, num_pairs)

### Example where pairs/edges are directed
```

```
# ground truth is a 3-star
true_pairs <- rbind(c(2,1), c(3,1), c(4,1))
# prediction gets direction of one edge incorrect
pred_pairs <- rbind(c(2,1), c(3,1), c(1,4))
# total number of pairs assuming 4 elements
num_pairs <- 4 * 4
eval_report_pairs(true_pairs, pred_pairs, num_pairs, ordered = TRUE)
```

`eval_report_clusters` *Evaluation Report for Clustering*

Description

Compute various evaluation measures for a predicted clustering using a ground truth clustering as a reference.

Usage

```
eval_report_clusters(true, pred)
```

Arguments

`true`	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
`pred`	predicted clustering represented as a membership vector.

Value

Returns a list containing the following measures:

homogeneity see `homogeneity`

completeness see `completeness`

v_measure see `v_measure`

rand_index see `rand_index`

adj_rand_index see `adj_rand_index`

variation_info see `variation_info`

mutual_info see `mutual_info`

fowlkes_mallows see `fowlkes_mallows`

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
eval_report_clusters(true, pred)
```

`eval_report_pairs` *Evaluation Report for Linked Pairs*

Description

Compute various evaluation measures for a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
eval_report_pairs(true_pairs, pred_pairs, num_pairs = NULL, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.
`num_pairs`	the total number of coreferent and non-coreferent pairs, excluding equivalent pairs with reversed ids. If not provided, measures that depend on the number of true negatives will be returned as `NA`.
`ordered`	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Value

Returns a list containing the following measures:

precision see `precision_pairs`

recall see `recall_pairs`

specificity see `specificity_pairs`

sensitivity see `sensitivity_pairs`

f1score see `f_measure_pairs`

accuracy see `accuracy_pairs`

balanced_accuracy see `balanced_accuracy_pairs`

fowlkes_mallows see `fowlkes_mallows_pairs`

See Also

The `contingency_table_pairs` function can be used to compute the contingency table for entity resolution or record linkage problems.

Examples

```
### Example where pairs/edges are undirected
# ground truth is 3-clique
true_pairs <- rbind(c(1,2), c(2,3), c(1,3))
# prediction misses one edge
pred_pairs <- rbind(c(1,2), c(2,3))
# total number of pairs assuming 3 elements
num_pairs <- 3 * (3 - 1) / 2
eval_report_pairs(true_pairs, pred_pairs, num_pairs)

### Example where pairs/edges are directed
# ground truth is a 3-star
true_pairs <- rbind(c(2,1), c(3,1), c(4,1))
# prediction gets direction of one edge incorrect
pred_pairs <- rbind(c(2,1), c(3,1), c(1,4))
# total number of pairs assuming 4 elements
num_pairs <- 4 * 4
eval_report_pairs(true_pairs, pred_pairs, num_pairs, ordered = TRUE)
```

---

fowlkes_mallows *Fowlkes-Mallows Index Between Clusterings*

Description

Computes the Fowlkes-Mallows index between two clusterings, such as a predicted and ground truth clustering.

Usage

```
fowlkes_mallows(true, pred)
```

Arguments

true	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
pred	predicted clustering represented as a membership vector.

Details

The Fowlkes-Mallows index is defined as the geometric mean of precision and recall, computed with respect to pairs of elements.

References

Fowlkes, E. B. and Mallows, C. L. "A Method for Comparing Two Hierarchical Clusterings." *Journal of the American Statistical Association* **78:383**, 553-569, (1983). doi:10.1080/01621459.1983.10478008

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
fowlkes_mallows(true, pred)
```

`fowlkes_mallows_pairs` *Fowlkes-Mallows Index of Linked Pairs*

Description

Computes the Fowlkes-Mallows index for a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
fowlkes_mallows_pairs(true_pairs, pred_pairs, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.
`ordered`	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The Fowlkes-Mallows index is defined as the geometric mean of precision P and recall R:

$$\sqrt{PR}.$$

References

Fowlkes, E. B. and Mallows, C. L. "A Method for Comparing Two Hierarchical Clusterings." *Journal of the American Statistical Association* **78:383**, 553-569, (1983). doi:10.1080/01621459.1983.10478008.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
fowlkes_mallows_pairs(true_pairs, pred_pairs, num_pairs)
```

`f_measure_pairs` *F-measure of Linked Pairs*

Description

Computes the F-measure (a.k.a. F-score) of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
f_measure_pairs(true_pairs, pred_pairs, beta = 1, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.
`beta`	non-negative weight. A value of 0 assigns no weight to recall (i.e. the measure reduces to precision), while larger values assign increasing weight to recall. A value of 1 weights precision and recall equally.
`ordered`	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The β-weighted F-measure is defined as the weighted harmonic mean of precision P and recall R:

$$(1 + \beta^2) \frac{P \cdot R}{\beta^2 \cdot P + R}.$$

References

Van Rijsbergen, C. J. "Information Retrieval." (2nd ed.). Butterworth-Heinemann, USA, (1979).

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
f_measure_pairs(true_pairs, pred_pairs, num_pairs)
```

`homogeneity` *Homogeneity Between Clusterings*

Description

Computes the homogeneity between two clusterings, such as a predicted and ground truth clustering.

Usage

```
homogeneity(true, pred)
```

Arguments

`true`	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
`pred`	predicted clustering represented as a membership vector.

Details

Homogeneity is an entropy-based measure of the similarity between two clusterings, say t and p. The homogeneity is high if clustering t only assigns members of a cluster to a single cluster in p. The homogeneity ranges between 0 and 1, where 1 indicates a perfect homogeneity.

References

Rosenberg, A. and Hirschberg, J. "V-measure: A conditional entropy-based external cluster evaluation measure." *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), (2007).

See Also

`completeness` evaluates the *completeness*, which is a dual measure to *homogeneity*. `v_measure` evaluates the harmonic mean of *completeness* and *homogeneity*.

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
homogeneity(true, pred)
```

mutual_info *Mutual Information Between Clusterings*

Description

Computes the mutual information between two clusterings, such as a predicted and ground truth clustering.

Usage

```
mutual_info(true, pred, base = exp(1))
```

Arguments

true	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
pred	predicted clustering represented as a membership vector.
base	base of the logarithm. Defaults to exp(1).

Details

Mutual information is an entropy-based measure of the similarity between two clusterings.

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
mutual_info(true, pred)
```

precision_pairs *Precision of Linked Pairs*

Description

Computes the precision of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
precision_pairs(true_pairs, pred_pairs, ordered = FALSE)
```

Arguments

true_pairs	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
pred_pairs	set of predicted coreferent pairs, following the same specification as true_pairs.
ordered	whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The precision is defined as:

$$\frac{|T \cap P|}{|P|}$$

where T is the set of true coreferent pairs and P is the set of predicted coreferent pairs.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
precision_pairs(true_pairs, pred_pairs, num_pairs)
```

---

rand_index *Rand Index Between Clusterings*

Description

Computes the Rand index (RI) between two clusterings, such as a predicted and ground truth clustering.

Usage

```
rand_index(true, pred)
```

Arguments

true	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
pred	predicted clustering represented as a membership vector.

Details

The Rand index (RI) can be expressed as:

$$\frac{a+b}{\binom{n}{2}}.$$

where

- n is the number of elements,
- a is the number of pairs of elements that appear in the same cluster in both clusterings, and
- b is the number of pairs of elements that appear in distinct clusters in both clusterings.

The RI takes on values between 0 and 1, where 1 denotes exact agreement between the clusterings and 0 denotes disagreement on all pairs of elements.

References

Rand, W. M. "Objective Criteria for the Evaluation of Clustering Methods." *Journal of the American Statistical Association* 66(336), 846-850 (1971). doi:10.1080/01621459.1971.10482356

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
rand_index(true, pred)
```

---

recall_pairs *Recall of Linked Pairs*

Description

Computes the precision of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
recall_pairs(true_pairs, pred_pairs, ordered = FALSE)

sensitivity_pairs(true_pairs, pred_pairs, ordered = FALSE)
```

Arguments

`true_pairs`	set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.
`pred_pairs`	set of predicted coreferent pairs, following the same specification as `true_pairs`.

ordered — whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The recall is defined as:

$$\frac{|T \cap P|}{|T|}$$

where T is the set of true coreferent pairs and P is the set of predicted coreferent pairs.

Note

`sensitivity_pairs` is an alias for `recall_pairs`.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
recall_pairs(true_pairs, pred_pairs, num_pairs)
```

---

specificity_pairs *Specificity of Linked Pairs*

Description

Computes the specificity of a set of *predicted* coreferent (linked) pairs given a set of *ground truth* coreferent pairs.

Usage

```
specificity_pairs(true_pairs, pred_pairs, num_pairs, ordered = FALSE)
```

Arguments

true_pairs — set of true coreferent pairs stored in a matrix or data.frame, where rows index pairs and columns index the ids of the constituents. Any pairs not included are assumed to be *non-coreferent*. Duplicate pairs (including equivalent pairs with reversed ids) are automatically removed.

pred_pairs — set of predicted coreferent pairs, following the same specification as `true_pairs`.

num_pairs — the total number of coreferent and non-coreferent pairs, excluding equivalent pairs with reversed ids.

ordered — whether to treat the element pairs as ordered—i.e. whether pair (x, y) is distinct from pair (y, x) for $x \neq y$. Defaults to FALSE, which is appropriate for clustering, undirected link prediction, record linkage etc.

Details

The specificity is defined as:

$$\frac{|P' \cap T'|}{|P'|}$$

where T' is the set of true non-coreferent pairs, P is the set of predicted non-coreferent pairs.

Examples

```
true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
pred_pairs <- rbind(c(1,2), c(2,3))         # prediction misses one edge
num_pairs <- 3                              # assuming 3 elements
specificity_pairs(true_pairs, pred_pairs, num_pairs)
```

---

variation_info *Variation of Information Between Clusterings*

Description

Computes the variation of information between two clusterings, such as a predicted and ground truth clustering.

Usage

```
variation_info(true, pred, base = exp(1))
```

Arguments

true	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
pred	predicted clustering represented as a membership vector.
base	base of the logarithm. Defaults to `exp(1)`.

Details

Variation of information is an entropy-based distance metric on the space of clusterings. It is unnormalized and varies between 0 and $\log(N)$ where N is the number of clustered elements. Larger values of the distance metric correspond to greater dissimilarity between the clusterings.

References

Arabie, P. and Boorman, S. A. "Multidimensional scaling of measures of distance between partitions." *Journal of Mathematical Psychology* **10:2**, 148-203, (1973). doi:10.1016/00222496(73)90012-6

Meilă, M. "Comparing Clusterings by the Variation of Information." In: Learning Theory and Kernel Machines, Lecture Notes in Computer Science **2777**, Springer, Berlin, Heidelberg, (2003). doi:10.1007/9783540451679_14

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
variation_info(true, pred)
```

`v_measure` *V-measure Between Clusterings*

Description

Computes the V-measure between two clusterings, such as a predicted and ground truth clustering.

Usage

```
v_measure(true, pred, beta = 1)
```

Arguments

`true`	ground truth clustering represented as a membership vector. Each entry corresponds to an element and the value identifies the assigned cluster. The specific values of the cluster identifiers are arbitrary.
`pred`	predicted clustering represented as a membership vector.
`beta`	non-negative weight. A value of 0 assigns no weight to completeness (i.e. the measure reduces to homogeneity), while larger values assign increasing weight to completeness. A value of 1 weights completeness and homogeneity equally.

Details

V-measure is defined as the β-weighted harmonic mean of homogeneity h and completeness c:

$$(1+\beta)\frac{h \cdot c}{\beta \cdot h + c}.$$

The range of V-measure is between 0 and 1, where 1 corresponds to a perfect match between the clusterings. It is equivalent to the normalised mutual information, when the aggregation function is the arithmetic mean.

References

Rosenberg, A. and Hirschberg, J. "V-measure: A conditional entropy-based external cluster evaluation measure." *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), (2007).

Becker, H. "Identification and characterization of events in social media." *PhD dissertation*, Columbia University, (2011).

See Also

homogeneity and completeness evaluate the component measures upon which this measure is based.

Examples

```
true <- c(1,1,1,2,2)  # ground truth clustering
pred <- c(1,1,2,2,2)  # predicted clustering
v_measure(true, pred)
```

Index