Package ‘clevr’

December 4, 2020

Type Package
Title Clustering and Link Prediction Evaluation in R
Version 0.1.1
Date 2020-11-29
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
LazyData true
Depends R (>= 3.0.2)
Imports Rcpp (>= 1.0.5), stats, Matrix
LinkingTo Rcpp, BH (>= 1.69.0)
RoxygenNote 7.1.1
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'
NeedsCompilation yes
Author Neil Marchant [aut, cre],
  Rebecca Steorts [aut]
Repository CRAN
Date/Publication 2020-12-04 09:20:05 UTC
R topics documented:

accuracy_pairs .......................................................... 2
adj_rand_index ............................................................ 3
balanced_accuracy_pairs ................................................. 4
canonicalize_pairs ......................................................... 5
clusters_to_membership .................................................... 6
completeness ................................................................. 8
contingency_table_clusters ............................................. 9
contingency_table_pairs .................................................. 9
eval_report_clusters ...................................................... 11
eval_report_pairs .......................................................... 12
fowlkes_mallows ............................................................ 13
fowlkes_mallows_pairs .................................................... 14
f_measure_pairs ............................................................ 15
homogeneity ................................................................. 16
mutual_info ................................................................. 17
precision_pairs ............................................................. 17
rand_index ................................................................. 18
recall_pairs ................................................................. 19
specificity_pairs ......................................................... 20
variation_info .............................................................. 21
v_measure ................................................................. 22

Index 24

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.
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:

\[
\frac{|T \cup P| + |T' \cup P'|}{N}
\]

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

```r
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**

```r
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:

\[
\text{ARI} = \frac{RI - E(RI)}{1 - E(RI)},
\]

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


Examples

```r
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

```r
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:

\[
\frac{|T \cup P|}{|P|} + \frac{|T' \cup P'|}{|P'|} \quad \frac{2}{2}
\]

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

```r
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

Description

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

Usage

```r
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 \((2, 3)\);
- 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.
**Example**

```r
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**

```r
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)
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 its 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 its name (if `members` is a named vector) or its integer index i.
chains to_membership

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


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 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 \( \text{pred} \) and cluster \( j \) in \( \text{true} \).

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)

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:

<table>
<thead>
<tr>
<th>Truth</th>
<th>Prediction</th>
<th>TRUE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>TP</td>
<td>FP</td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>FN</td>
<td>TN</td>
<td></td>
</tr>
</tbody>
</table>

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

```r
# 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)
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

```r
### 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**

```r
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**
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


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_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 \(\beta\)-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


Examples

```r
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**


**See Also**

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

**Examples**

```r
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**

```r
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**

```r
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**

```r
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:

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

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

Examples

```r
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 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**


**Examples**

```r
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**

```r
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`. 
specificity_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:

\[
\frac{|T \cup P|}{|T|}
\]

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)

specficity_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.
um_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:

\[
\frac{|P' \cup T'|}{|P'|}
\]

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

Examples

```r
t_true_pairs <- rbind(c(1,2), c(2,3), c(1,3)) # ground truth is 3-clique
t_pred_pairs <- rbind(c(1,2), c(2,3))       # prediction misses one edge
t_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

```r
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 un-normalized 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


Examples

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

<table>
<thead>
<tr>
<th>v_measure</th>
<th>$V$-measure Between Clusterings</th>
</tr>
</thead>
</table>

Description

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

Usage

```r
v_measure(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

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

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

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


See Also

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

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

accuracy_pairs, 2, 12

adj_rand_index, 3, 11

balanced_accuracy_pairs, 4, 12

canonicalize_pairs, 5

clusters_to_membership, 6

clusters_to_pairs, 10

clusters_to_pairs
  (clusters_to_membership), 6

completeness, 8, 11, 16, 22

contingency_table_clusters, 9

contingency_table_pairs, 9, 12

eval_report_clusters, 9, 11

eval_report_pairs, 10, 12

f_measure_pairs, 12, 15

fowlkes_mallows, 11, 13

fowlkes_mallows_pairs, 12, 14

homogeneity, 8, 11, 16, 22

membership_to_clusters
  (clusters_to_membership), 6

membership_to_pairs, 10

membership_to_pairs
  (clusters_to_membership), 6

mutual_info, 11, 17

pairs_to_clusters
  (clusters_to_membership), 6

pairs_to_membership
  (clusters_to_membership), 6

precision_pairs, 12, 17

rand_index, 11, 18

recall_pairs, 12, 19

sensitivity_pairs, 12

sensitivity_pairs (recall_pairs), 19

specificity_pairs, 12, 20

v_measure, 8, 11, 16, 22

variation_info, 11, 21