Package ‘mclustcomp’

June 13, 2021

Type Package
Title Measures for Comparing Clusters
Version 0.3.3
Description Given a set of data points, a clustering is defined as a disjoint partition
where each pair of sets in a partition has no overlapping elements.
This package provides 25 methods that play a role somewhat similar to
distance or metric that measures similarity of two clusterings - or partitions.
License GPL (>= 3)
Encoding UTF-8
Imports Rcpp, Rdpack
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 7.1.1
RdMacros Rdpack
NeedsCompilation yes
Author Kisung You [aut, cre]
Maintainer Kisung You <kisungyou@outlook.com>
Repository CRAN
Date/Publication 2021-06-13 04:40:11 UTC

R topics documented:
mclustcomp-package ..................................................... 2
mclustcomp ............................................................... 2
Index 6
mclustcomp-package  Measures for Comparing Clusterings

Description

Given a set of data points \( D \), a clustering \( C = (C_1, C_2, ..., C_k) \) is a partition where each pair of sets \( C_i \) and \( C_j \) has no overlapping elements. mclustcomp package provides a collection of methods that play a role similar to distance or metric in that measures similarity of two clusterings (or, partitions) \( C \) and \( C' \). For a more detailed description, see Meila, M. (2005) <doi:10.1145/1102351.1102424>.

Usage

\[
mclustcomp(x, y, types = "all", tversky.param = list())
\]

Arguments

- **x**, **y**: vectors of clustering labels
- **types**: "all" for returning scores for every available measure. Either a single score name or a vector of score names can be supplied. See the section for the list of the methods for details.
- **tversky.param**: a list of parameters for Tversky index; alpha and beta for weight parameters, and sym, a logical where FALSE stands for original method, TRUE for a revised variant to symmetrize the score. Default (alpha,beta)=(1,1).

Value

a data frame with columns types and corresponding scores.
Category 1. Counting Pairs

<table>
<thead>
<tr>
<th>TYPE</th>
<th>FULL NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>'adjrand'</td>
<td>Adjusted Rand index.</td>
</tr>
<tr>
<td>'chisq'</td>
<td>Chi-Squared Coefficient.</td>
</tr>
<tr>
<td>'fmi'</td>
<td>Fowlkes-Mallows index.</td>
</tr>
<tr>
<td>'jaccard'</td>
<td>Jaccard index.</td>
</tr>
<tr>
<td>'mirkin'</td>
<td>Mirkin Metric, or Equivalence Mismatch Distance.</td>
</tr>
<tr>
<td>'overlap'</td>
<td>Overlap Coefficient, or Szymkiewicz-Simpson coefficient.</td>
</tr>
<tr>
<td>'pd'</td>
<td>Partition Difference.</td>
</tr>
<tr>
<td>'rand'</td>
<td>Rand Index.</td>
</tr>
<tr>
<td>'sdc'</td>
<td>Sørensen–Dice Coefficient.</td>
</tr>
<tr>
<td>'smc'</td>
<td>Simple Matching Coefficient.</td>
</tr>
<tr>
<td>'tanimoto'</td>
<td>Tanimoto index.</td>
</tr>
<tr>
<td>'tversky'</td>
<td>Tversky index.</td>
</tr>
<tr>
<td>'wallace1'</td>
<td>Wallace Criterion Type 1.</td>
</tr>
<tr>
<td>'wallace2'</td>
<td>Wallace Criterion Type 2.</td>
</tr>
</tbody>
</table>

Note that Tanimoto Coefficient and Dice’s coefficient are special cases with (alpha,beta) = (1,1) and (0.5,0.5), respectively.

Category 2. Set Overlaps/Matching

<table>
<thead>
<tr>
<th>TYPE</th>
<th>FULL NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>'f'</td>
<td>F-Measure.</td>
</tr>
<tr>
<td>'mhm'</td>
<td>Meila-Heckerman Measure.</td>
</tr>
<tr>
<td>'mmm'</td>
<td>Maximum-Match Measure.</td>
</tr>
<tr>
<td>'vdm'</td>
<td>Van Dongen Measure.</td>
</tr>
</tbody>
</table>

Category 3. Information Theory

<table>
<thead>
<tr>
<th>TYPE</th>
<th>FULL NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>'jent'</td>
<td>Joint Entropy</td>
</tr>
<tr>
<td>'mi'</td>
<td>Mutual Information.</td>
</tr>
<tr>
<td>'nmi1'</td>
<td>Normalized Mutual Information by Strehl and Ghosh.</td>
</tr>
<tr>
<td>'nmi2'</td>
<td>Normalized Mutual Information by Fred and Jain.</td>
</tr>
<tr>
<td>'nmi3'</td>
<td>Normalized Mutual Information by Danon et al.</td>
</tr>
<tr>
<td>'nvi'</td>
<td>Normalized Variation of Information.</td>
</tr>
<tr>
<td>'vi'</td>
<td>Variation of Information.</td>
</tr>
</tbody>
</table>
References


Examples

```r
## example 1. compare two identical clusterings
x = sample(1:5,20,replace=TRUE) # label from 1 to 5, 10 elements
ty = x # set two labels x and y equal
mclustcomp(x,y) # show all results

## example 2. selection of a few methods
z = sample(1:4,20,replace=TRUE) # generate a non-trivial clustering
cmethods = c("jaccard","tanimoto","rand") # select 3 methods
mclustcomp(x,z,types=cmethods) # test with the selected scores

## example 3. tversky.param
tparam = list() # create an empty list
tparam$alpha = 2
tparam$beta = 3
tparam$sym = TRUE
mclustcomp(x,z,types="tversky") # default set as Tanimoto case.
mclustcomp(x,z,types="tversky",tversky.param=tparam)
```
Index

mclustcomp, 2
mclustcomp-package, 2