Package 'aricode'

July 22, 2025

Type Package

Title Efficient Computations of Standard Clustering Comparison Measures

Version 1.0.3

Maintainer Julien Chiquet <julien.chiquet@inrae.fr>

Description Implements an efficient O(n) algorithm based on bucket-sorting for fast computation of standard clustering comparison measures. Available measures include adjusted Rand index (ARI), normalized information distance (NID), normalized mutual information (NMI), adjusted mutual information (AMI), normalized variation information (NVI) and entropy, as described in Vinh et al (2009) <doi:10.1145/1553374.1553511>. Include AMI (Adjusted Mutual Information) since version 0.1.2,
a modified version of ARI (MARI), as described in Sundqvist et al. <doi:10.1007/s00180-022-01230-7>
and simple Chi-square distance since version 1.0.0.

License GPL (>= 3)

URL https://github.com/jchiquet/aricode

BugReports https://github.com/jchiquet/aricode/issues

Encoding UTF-8

Imports Matrix, Rcpp

Suggests testthat, spelling

LinkingTo Rcpp

RoxygenNote 7.2.3

Language en-US

NeedsCompilation yes

Author Julien Chiquet [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3629-3429>),
Guillem Rigaill [aut],
Martina Sundqvist [aut],
Valentin Dervieux [ctb],
Florent Bersani [ctb]

Repository CRAN

Date/Publication 2023-10-20 15:10:02 UTC

Contents

`AMI` *Adjusted Mutual Information*

Description

A function to compute the adjusted mutual information between two classifications

Usage

```
AMI(c1, c2)
```

Arguments

`c1`	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
`c2`	a vector containing the labels of the second classification.

Value

a scalar with the adjusted rand index.

See Also

`ARI`, `RI`, `NID`, `NVI`, `NMI`, `clustComp`

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
AMI(cl,iris$Species)
```

ARI *Adjusted Rand Index*

Description

A function to compute the adjusted rand index between two classifications

Usage

```
ARI(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the adjusted rand index.

See Also

RI, NID, NVI, NMI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
ARI(cl,iris$Species)
```

Chi2 *Chi-square statistics*

Description

A function to compute the Chi-2 statistics

Usage

```
Chi2(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the chi-square statistics.

See Also

ARI, NID, NVI, NMI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
Chi2(cl,iris$Species)
```

clustComp *Measures of similarity between two classification*

Description

A function various measures of similarity between two classifications

Usage

```
clustComp(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a list with the RI, ARI, NMI, NVI and NID.

See Also

RI, NID, NVI, NMI, ARI

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
clustComp(cl,iris$Species)
```

`entropy` *Entropy*

Description

A function to compute the empirical entropy for two vectors of classification and the joint entropy

Usage

```
entropy(c1, c2)
```

Arguments

`c1`	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
`c2`	a vector containing the labels of the second classification.

Value

a list with the two conditional entropies, the joint entropy and output of sortPairs.

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
entropy(cl,iris$Species)
```

MARI *Modified Adjusted Rand Index*

Description

A function to compute a modified adjusted rand index between two classifications as proposed by Sundqvist et al. in prep, based on a multinomial model.

Usage

```
MARI(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the modified ARI.

See Also

ARI, NID, NVI, NMI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
MARI(cl,iris$Species)
```

MARIraw *raw Modified Adjusted Rand Index*

Description

A function to compute a modified adjusted rand index between two classifications as proposed by Sundqvist et al. in prep, based on a multinomial model. Raw means, that the index is not divided by the (maximum - expected) value.

Usage

```
MARIraw(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the modified ARI without the division by the (maximum - expected)

See Also

ARI, NID, NVI, NMI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
MARIraw(cl,iris$Species)
```

NID *Normalized information distance (NID)*

Description

A function to compute the NID between two classifications

Usage

```
NID(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the normalized information distance .

See Also

RI, NMI, NVI, ARI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
NID(cl,iris$Species)
```

NMI *Normalized mutual information (NMI)*

Description

A function to compute the NMI between two classifications

Usage

```
NMI(c1, c2, variant = c("max", "min", "sqrt", "sum", "joint"))
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.
variant	a string in ("max", "min", "sqrt", "sum", "joint"): different variants of NMI. Default use "max".

Value

a scalar with the normalized mutual information .

See Also

RI, NID, NVI, ARI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
NMI(cl,iris$Species)
```

NVI *Normalized variation of information (NVI)*

Description

A function to compute the NVI between two classifications

Usage

```
NVI(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the normalized variation of information.

See Also

RI, NID, NMI, ARI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
NVI(cl,iris$Species)
```

RI *Rand Index*

Description

A function to compute the rand index between two classifications

Usage

```
RI(c1, c2)
```

Arguments

c1	a vector containing the labels of the first classification. Must be a vector of characters, integers, numerics, or a factor, but not a list.
c2	a vector containing the labels of the second classification.

Value

a scalar with the rand index.

See Also

ARI, NID, NVI, NMI, clustComp

Examples

```
data(iris)
cl <- cutree(hclust(dist(iris[,-5])), 4)
RI(cl,iris$Species)
```

`sortPairs` *Sort Pairs*

Description

A function to sort pairs of integers or factors and identify the pairs

Usage

```
sortPairs(c1, c2, spMat = FALSE)
```

Arguments

`c1`	a vector of length n with value between 0 and N1 < n
`c2`	a vector of length n with value between 0 and N2 < n
`spMat`	logical: send back the contingency table as sparsely encoded (cost more than the algorithm itself). Default is FALSE

Index