

Package ‘validator’

February 15, 2012

Type Package

Title External and Internal Validation Indices

Version 1.0

Date 2010-09-04

Author Marcus Scherl

Depends clv, clValid, flexclust, mlbench

Maintainer Marcus Scherl <marcus.scherl@gmx.net>

Description This package contains external and internal validation indices. The internal validation functions are mainly implemented in ‘cclust’ and revised with small changes. The external validation functions are using the similarity table from the package ‘clv’ and further external indices are included in this package.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2010-09-05 06:48:24

R topics documented:

extVal	2
intVal	3
Index	6

extVal

*External Validation Indices***Description**

This function is calculating the values of certain external validation indices. The values compare the predicted cluster assignment with their true cluster labelling existing in the data.

Usage

```
extVal(x, y, index = "all")
```

Arguments

x Vector of the true cluster assignment

y Vector of the cluster assignment, which has to be compare with the true one

index The external indices, which are calculated: "Hamann", "Czekanowski", "Kulczynski", "McConnaughey", "Peirce", "Wallace1", "Wallace2", "Gamma", "Sokal1", "Fager", "Sokal2", "Sokal3", "Gower", "Roger", "Kruskal", "Pearson", "Rand", "Jaccard", "Folkes", "Russel", and "all".

Details

The data points are counted in a pairwise co-assignment. Given two partitions named C1 and C2, the quantities a, b, c and d are computed for the pairs of the data points x_i and x_j and their cluster assignments. The numbers a and d are counted as the agreements between the two cluster partitions C1 and C2, whereas b and c are the disagreements of these two partitions:

- a - number of pairs in the same cluster of both partitions,
- b - number of pairs, which are in the same cluster of partition C1 but not of partition C2,
- c - number of pairs, in which both data points are in different clusters in the partitions C1 and C2,
- d - number of pairs in different clusters of both partitions.

Then, the external indices are computed with these values:

Rand	$\frac{a+d}{a+b+c+d}$
Hamann	$\frac{(a+d)-(b+c)}{a+b+c+d}$
Czekanowski	$\frac{2a}{2a+b+c}$
Kulczynski	$\frac{1}{2} \left(\frac{a}{a+b} + \frac{a}{a+c} \right)$
McConnaughey	$\frac{a^2-bc}{(a+b)(a+c)}$
Peirce	$\frac{ad-bc}{(a+c)-(b+d)}$
Fowlkes	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Wallace1	$\frac{a}{a+b}$
Wallace2	$\frac{a}{a+c}$
Gamma	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$

Sokal1	$\frac{1}{4} \left(\frac{a}{a+b} + \frac{a}{a+c} + \frac{d}{d+b} + \frac{d}{d+c} \right)$
Russel	$\frac{a}{a+b+c+d}$
Fager	$\frac{\frac{a}{\sqrt{(a+b)(a+c)}}}{2\sqrt{(a+b)}} - \frac{1}{2\sqrt{(a+b)}}$
Pearson	$\frac{ad-bc}{(a+b)(a+c)(c+d)(b+d)}$
Jaccard	$\frac{a}{a+b+c}$
Sokal2	$\frac{a}{a+2(b+c)}$
Sokal3	$\frac{ad}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$
Gower	$\frac{a+d}{a+\frac{1}{2}(b+c)+d}$
Rogers	$\frac{a+d}{a+2(b+c)+d}$
Kruskal	$\frac{ad-bc}{ad+bc}$

Value

This function returns a vector with the external validation indices.

Author(s)

Marcus Scherl

References

Ahmed N Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko *On similarity indices and correction for chance agreement*.

Examples

```
#require(mlbench)
#require(flexclust)

x <- mlbench.2dnormals(500, 3)
cl <- kcca(x$x, 3)
pred <- predict(cl, x$x)
extVal(pred, x$class)
```

intVal

Internal Validation Indices

Description

This function is calculating the values of certain internal validation indices.

Usage

```
intVal(y, x, index = "all")
```

Arguments

y	Object of class <code>kcca</code> returned by clustering methods from the package <code>flexclust</code>
x	Data matrix, which contains the observations of clustering or data matrix of a test data set
index	The internal validation indices, which are calculated: "calinski", "db", "hartigan", "ratkowsky", "scott", "marriot", "ball", "trcovw", "tracew", "friedman", "rubin", "xuindex", "dunn", "connectivity", "silhouette", and "all".

Details

The internal validation indices based all on the references below. The indices are defined as:

calinski: $\frac{SSB/(k-1)}{SSW/(n-k)}$, where *SSB* is the sum of squares between, *SSW* is the sum of squares between, *n* is the number of data points, and *k* is the number of clusters.

db: $\frac{1}{k} \sum_{i=1}^k R_i$, where R_i as the maximum value of $R_{ij} = \frac{s_i + s_j}{d_{ij}}$ with s_i is the similarity in the clusters and d_{ij} is the dissimilarity between the two clusters.

hartigan: $\log \frac{SSB}{SSW}$

ratkowsky: \tilde{c}/\sqrt{k} , where $\tilde{c} = \text{mean} \sqrt{\text{var}SSB/\text{var}SST}$ and the abbreviation *emphvar* stands for each variable and *SST* for the sum of squares total.

ball: $\frac{SSW}{k}$

xuindex: $d \log(\sqrt{SSW/(dn^2)}) + \log(k)$, with *d* as the dimension of the data points.

scott: $n \log \frac{\det(T)}{\det(W)}$, with *T* is the scatter distance matrix and *W* is the pooled within groups scatter matrix.

marriot: $k^2 \det(W)$

trcovw: $\text{trace}(\text{cov}W)$

tracew: $\text{trace}(W)$

friedman: $\text{trace}(W^{-1}B)$, where *B* is the between groups scatter matrix.

rubin: $\det(T)/\det(W)$

The R code of these functions above are taken from the package `cclust` with small changes, e.g. by computing the *SSW* regarding their cluster centers.

dunn: $D(C) = \frac{\min_{C_k, C_l \in C, C_k \neq C_l} \left(\min_{x_i \in C_k, x_j \in C_l} \text{dist}(x_i, x_j) \right)}{\max_{C_m \in C} \text{diam}(C_m)}$, where $\text{diam}(C_m)$ is maximum distance

between each data item in the cluster C_m , and $\text{dist}(x_i, x_j)$ is the distance between the pairs of the data points.

silhouette: $S(x_i) = \frac{b_i - a_i}{\max(a_i, b_i)}$, where a_i is the average distance between the data points to all the other observations in the same cluster and b_i is the average distance between data points to all other points from the closest neighbouring cluster. Then, the average of all Silhouette Widths is computed.

connectivity: $\text{Conn}(C) = \sum_{i=1}^N \sum_{j=1}^L x_{i, nn_i(j)}$, where $nn_i(j)$ be the *j*-th neighbour of the data point x_i ; so $x_{i, nn_i(j)}$ is zero if x_i and $nn_i(j)$ are in the same cluster, otherwise the value is computed by $\frac{1}{j}$ and *L* determines the number of neighbours that contribute to the connectivity measure.

Value

This function returns a vector with the internal validation indices.

Author(s)

Marcus Scherl

References

Glenn W. Milligan and Martha C. Cooper, *An examination of procedures for determining the number of clusters in a dataset.*

Julia Handl, Joshua Knowles, and Douglas B. Kell, *Computational cluster validation in post-genomic data analysis*, <http://dbkgroup.org/handl/clustervalidation/>.

Andreas Weingessel, Evgenia Dimitriadou, and Sara Dolnicar, *An examination of indexes for determining the number of clusters in binary data sets.*

Guy Brock, Vasyi Pihur, Susmita Datta, and Somnath Datta, *clValid: An R package for cluster validation*, <http://www.jstatsoft.org/v25/i04>.

Examples

```
# require(mlbench)
# require(flexclust)

x <- mlbench.2dnormals(500, 3)
cl <- kcca(x$x, 3)
intVal(cl, x$x)

x <- mlbench.2dnormals(500, 3)
cl <- kmeans(x$x, 3)
cl <- as.kcca(cl, x$x)
intVal(cl, x$x)
```

Index

*Topic **extVal**

extVal, 2

*Topic **intVal**

intVal, 3

extVal, 2

intVal, 3