Package ‘PaolaR6’

October 30, 2023

Type Package

Title Confusion Matrix

Version 0.1.0

Author Paola

Maintainer Paola Barba Ceballos <pbarba@ujaen.es>

Description This package provides useful functions for the analysis of confusion matrices in classification problems. Includes methods to calculate overall accuracy, user accuracy, and map creator accuracy.

Imports R6

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

R topics documented:

MatCon *Confusion matrix*

Description

Using the confusion matrix, various indices are calculated.

Value

Object of class MatCon #y mas

Methods

Public methods:

- `MatCon$new()`
- `MatCon$oa()`
- `MatCon$dec()`
- `MatCon$ua()`
- `MatCon$uai()`
- `MatCon$pai()`
- `MatCon$pa()`
- `MatCon$aup()`
- `MatCon$ICSI()`
- `MatCon$mah()`
- `MatCon$mas()`
- `MatCon$cku()`
- `MatCon$ckp()`
- `MatCon$mcku()`
- `MatCon$mckp()`
- `MatCon$ecnu()`
- `MatCon$ecnp()`
- `MatCon$aau()`
- `MatCon$Normalize()`
- `MatCon$clone()`

Method `new()`**:**

Usage:

```
MatCon$new(values, ID = NULL, Date = NULL)
```

Arguments:

`values` Confusion matrix

`ID` Identifier (optional)

`Date` System or user-provided date (optional)

Method `oa()`**:** Overall accuracy for a particular classified image/map is then calculated by dividing the sum of the entries that form the major diagonal (i.e., the number of correct classifications) by the total number of samples taken.
The mathematical expression is:

$$oa = \frac{\sum_{i=1}^{n} x_{ii}}{\sum_{i,j=1}^{n} x_{ij}}$$

Where:

1. ‘oa‘: overall accuracy.
2. ‘x_ii‘: diagonal element of the matrix.
3. ‘x_ij‘: element of the matrix.

This represents a mathematical expression with a fraction.

Usage:

```
MatCon$oa(...)
```

Arguments:

... (ignored).

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$oa()
```

Method `dec()`**:** Determines whether a value is decimal or not.

Usage:

```
MatCon$dec(...)
```

Arguments:

... (ignored).

Returns: TRUE or FALSE

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$dec()
```

Method `ua()`**:** The accuracy from the point of view of a map user, not the map maker. The mathematical expression is:

$$ua = \frac{x_{ii}}{\sum_{j=1}^{n} x_{ij}}$$

where:

1. ‘ua‘: user accuracy.
2. ‘x_ii‘: diagonal element of the matrix.
3. ‘x_ij‘: element of the matrix.

Usage:

```
MatCon$ua(...)
```

Arguments:

... (ignored).

Returns: vector of values with the user’s accuracy indexes of all classes

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A,ID=1,Date="30/10/2023")
p$ua()
```

Method `uai()`**:** The accuracy from the point of view of a map user, not the map maker.

$$ua_i = \frac{x_{ii}}{\sum_{j=1}^{n} x_{ij}}$$

where:

1. ‘ua_i‘: user accuracy.
2. ‘x_ii‘: diagonal element of the matrix.
3. ‘x_ij‘: element of the matrix.

Usage:

```
MatCon$uai(i)
```

Arguments:

`i` User class to evaluate

Returns: Class i user accuracy index

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$uai(2)
```

Method `pai()`**:** The map accuracy from the point of view of the map maker (the producer).

$$pa_i = \frac{x_{jj}}{\sum_{j=1}^{n} x_{ij}}$$

where:

1. ‘pa_i‘: producer accuracy.
2. ‘x_jj‘: diagonal element of the matrix.
3. ‘x_ij‘: element of the matrix.

Usage:

```
MatCon$pai(i)
```

Arguments:

`i` Producer class to evaluate

Returns: Class i producer accuracy index

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$pai(1)
```

Method `pa()`**:** The map accuracy from the point of view of the map maker (the producer).

$$pa = \frac{x_{jj}}{\sum_{j=1}^{n} x_{ij}}$$

where:

1. ‘pa‘: producer accuracy.
2. ‘x_jj‘: diagonal element of the matrix.
3. ‘x_ij‘: element of the matrix.

Usage:

```
MatCon$pa(...)
```

Arguments:

`...` (ignored).

Returns: Vector of values with the producer’s accuracy indexes of all classes

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$pa()
```

Method `aup()`**:** Average of the accuracy from the point of view of a map user, not the map maker and the map accuracy from the point of view of the map maker (the producer).

$$aup = \frac{ua_i + pa_i}{2}$$

where:

1. ‘aup‘: Average of user’s and producer’s accuracy.
2. ‘ua_i‘: user accuracy
3. ‘pa_i‘: producer accuracy.

Usage:

```
MatCon$aup(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$aup(2)
```

Method `ICSI()`**:** The Individual Classification Success Index (ICSI) applies to the classification effectiveness for one particular class of interest.

$$ICSI = ua_i + pa_i - 1$$

where:

1. ‘ICSI‘: Individual classification success index.
2. ‘ua_i‘: user accuracy.
3. ‘pa_i‘: producer accuracy.

Usage:

```
MatCon$ICSI(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ICSI(2)
```

Method `mah()`**:** The probability that a randomly chosen point of a specific class on the map has a correspondence of the same class in the same position in the field and that a randomly chosen point in the field of the same class has a correspondence of the same class in the same position on the map.

$$mah = \frac{2}{\frac{1}{ua_i} + \frac{1}{pa_i}}$$

where:

1. ‘mah‘: Hellden’s mean accuracy.
2. ‘ua_i‘: user accuracy.
3. ‘pa_i‘: producer accuracy.

Usage:

```
MatCon$mah(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mah(2)
```

Method `mas()`**:** Mapping accuracy for each class is stated as the number of correctly classified pixels (equal to the total in the correctly classified area) in terms of all pixels affected by its classification (equal to this total in the displayed area as well as the pixels involved in errors of commission and omission).

$$mas = \frac{x_{ii}}{\sum_{j=1}^{n} x_{\cdot j} + \sum_{i=1}^{n} x_{i\cdot} - x_{ii}}$$

where:

1. ‘mas‘: Short’s mapping accuracy
2. ‘x_ii‘: diagonal element of the matrix.
3. ‘x_.j‘: sum with respect to j (rows).
4. ‘x_i.‘: sum with respect to i (columns).

Usage:

```
MatCon$mas(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mas(2)
```

Method `cku()`**:** Conditional Kappa will identify the degree of agreement between the two raters for each possible category.

$$cku = \frac{ua_i - \frac{\sum_{i=1}^{n} x_{i\cdot}}{\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}}}{1 - \frac{\sum_{i=1}^{n} x_{i\cdot}}{\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}}}$$

where:

1. ‘cku‘: Conditional kappa (user’s).
2. ‘ua_i‘: user accuracy.
3. ‘x_ii‘: diagonal element of the matrix.
4. ‘x_.j‘: sum with respect to j (rows).
5. ‘x_i.‘: sum with respect to i (columns).

Usage:

```
MatCon$cku(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$cku(2)
```

Method `ckp()`**:** Conditional Kappa will identify the degree of agreement between the two raters for each possible category.

$$ckp = \frac{pa_i - \frac{\sum_{j=1}^{n} x_{\cdot j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}}}{1 - \frac{\sum_{j=1}^{n} x_{\cdot j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}}}$$

where:

1. ‘ckp‘: Conditional kappa (producer’s).
2. ‘pa_i‘: producer accuracy.
3. ‘x_ii‘: diagonal element of the matrix.
4. ‘x_.j‘: sum with respect to j (rows).
5. ‘x_i.‘: sum with respect to i (columns).

Usage:

```
MatCon$ckp(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ckp(2)
```

Method `mcku()`**:** Modified kappa index for the user

$$mcku = \frac{ua_i - \frac{1}{\sqrt{card(p)}}}{1 - \frac{1}{\sqrt{card(p)}}}$$

where:

1. ‘mcku‘: Modified conditional kappa (user’s).
2. ‘ua_i‘: user accuracy.
3. ‘card(p)‘: number of elements of the matrix, cardinal of the matrix.

Usage:

```
MatCon$mcku(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mcku(2)
```

Method `mckp()`**:** Modified kappa index for the producer

$$mckp = \frac{pa_i - \frac{1}{\sqrt{card(p)}}}{1 - \frac{1}{\sqrt{card(p)}}}$$

where:

1. ‘mckp‘: Modified conditional kappa (producer’s).
2. ‘pa_i‘: producer accuracy.
3. ‘card(p)‘: number of elements of the matrix, cardinal of the matrix.

Usage:

```
MatCon$mckp(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mckp(2)
```

Method `ecnu()`**:** Relative entropy is a quantity that measures the difference between two maps.

$$H(A) = -\sum_{j=1}^{n}((\frac{\sum_{i=1}^{n} x_{i\cdot}}{\sum_{i,j=1}^{n} x_{ij}})\cdot\log(\frac{\sum_{i=1}^{n} x_{i\cdot}}{\sum_{i,j=1}^{n} x_{ij}}))H(A|b_i) = -\sum_{j=1}^{n}((\frac{x_{ij}}{\sum_{j=1}^{n} x_{\cdot j}})\cdot\log(\frac{x_{ij}}{\sum_{j=1}^{n} x_{\cdot j}}))ecnu = \frac{H(A) - H(}{H(A)}$$

where:

1. ‘ecnu‘: Relative change of entropy given a category on map.
2. ‘H(A)‘: the entropy of the map.
3. ‘x_.j‘: sum with respect to j (rows).
4. ‘x_i.‘: sum with respect to i (columns).
5. ‘H(A|b_i)‘: Entropy of map A knowing that the location corresponding to map B is in class b_i.

Usage:

```
MatCon$ecnu(i)
```

Arguments:

`i` Class to evaluate.

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ecnu(2)
```

Method `ecnp()`**:** Relative entropy is a quantity that measures the difference between two ground truthing.

$$H(B) = -\sum_{i=1}^{n}((\frac{\sum_{j=1}^{n} x_{\cdot j}}{\sum_{i,j=1}^{n} x_{ij}})\cdot\log(\frac{\sum_{j=1}^{n} x_{\cdot j}}{\sum_{i,j=1}^{n} x_{ij}}))H(B|a_j) = -\sum_{j=1}^{n}((\frac{x_{ij}}{\sum_{i=1}^{n} x_{i\cdot}})\cdot\log(\frac{x_{ij}}{\sum_{i=1}^{n} x_{i\cdot}}))ecnp = \frac{H(B) - H(}{H(B)}$$

where:

1. ‘ecnp‘: Relative change of entropy given a category on ground truthing.
2. ‘H(B)‘: the entropy of the map.
3. ‘x_.j‘: sum with respect to j (rows).
4. ‘x_i.‘: sum with respect to i (columns).
5. ‘H(B|a_j)‘: Entropy of map B knowing that the location corresponding to map A is in class a_j.

Usage:

```
MatCon$ecnp(i)
```

Arguments:

`i` Class to evaluate

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ecnp(2)
```

Method `aau()`**:** The average accuracy is an average of the accuracy of individual categories. Because the individual categories can be the user’s or the producer’s accuracy, it can be computed in both ways accordingly.

$$aau = \frac{1}{\sqrt{card(p)}} \sum_{i=1}^{n} \frac{x_{ii}}{\sum_{j=1}^{n} x_{ij}}$$

where:

1. ‘aau‘: Average accuracy from user’s perspective
2. ‘card(p)‘: number of elements of the matrix, cardinal of the matrix.
3. ‘x_.j‘: sum with respect to j (rows).
4. ‘x_ii‘: diagonal element of the matrix.

Usage:

```
MatCon$aau()
```

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$aau()
```

Method `Normalize()`**:** An iterative process is carried out where each element is divided by the total of the sum of its row, thus obtaining new values. In the next iteration, all the elements are added by columns and each element is divided by the total of its column and they obtain new values, and so on.

Usage:

```
MatCon$Normalize(n)
```

Arguments:

`n` Iteration

Examples:

```
A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$Normalize(100)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
MatCon$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

References

Story, M., & Congalton, R. G. (1986). Accuracy assessment: a user's perspective. Photogrammetric Engineering and remote sensing, 52(3), 397-399.

Koukoulas, S., & Blackburn, G. A. (2001). Introducing new indices for accuracy evaluation of classified images representing semi-natural woodland environments. Photogrammetric Engineering and Remote Sensing, 67(4), 499-510.

Turk, G. (2002). Map evaluation and" chance correction". Photogrammetric Engineering and Remote Sensing, 68(2), 123-129.

Helldén, U. (1980). A test of landsat-2 imagery and digital data for thematic mapping illustrated by an environmental study in northern Kenya, Lund University. Natural Geography Institute Report No. 47.

Rosenfield, G. H., & Fitzpatrick-Lins, K. (1986). A coefficient of agreement as a measure of thematic classification accuracy. Photogrammetric engineering and remote sensing, 52(2), 223-227.

Short, N. M. (1982). The Landsat tutorial workbook: Basics of satellite remote sensing (Vol. 1078). National Aeronautics and Space Administration, Scientific and Technical Information Branch.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. Remote sensing of Environment, 62(1), 77-89.

Finn, J. T. (1993). Use of the average mutual information index in evaluating classification error and consistency. International Journal of Geographical Information Science, 7(4), 349-366.

Tung, F., & LeDrew, E. (1988). The determination of optimal threshold levels for change detection using various accuracy indexes. Photogrammetric Engineering and Remote Sensing, 54(10), 1449-1454.

Fienberg, S. E. (1970). An iterative procedure for estimation in contingency tables. The Annals of Mathematical Statistics, 41(3), 907-917.

Examples

```
C = matrix( c(5, 0, 1, 0,4,0,0,0,3), nrow=3,  ncol=3)
mc2 <- MatCon$new (C,ID=5,Date="27-10-2023")

## ------------------------------------------------
## Method `MatCon$oa`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$oa()

## ------------------------------------------------
## Method `MatCon$dec`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
```

```
p<-MatCon$new(A)
p$dec()

## ------------------------------------------------
## Method `MatCon$ua`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A,ID=1,Date="30/10/2023")
p$ua()

## ------------------------------------------------
## Method `MatCon$uai`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$uai(2)

## ------------------------------------------------
## Method `MatCon$pai`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$pai(1)

## ------------------------------------------------
## Method `MatCon$pa`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$pa()

## ------------------------------------------------
## Method `MatCon$aup`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$aup(2)

## ------------------------------------------------
## Method `MatCon$ICSI`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ICSI(2)

## ------------------------------------------------
## Method `MatCon$mah`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
```

```
p$mah(2)

## ------------------------------------------------
## Method `MatCon$mas`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mas(2)

## ------------------------------------------------
## Method `MatCon$cku`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$cku(2)

## ------------------------------------------------
## Method `MatCon$ckp`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ckp(2)

## ------------------------------------------------
## Method `MatCon$mcku`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mcku(2)

## ------------------------------------------------
## Method `MatCon$mckp`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$mckp(2)

## ------------------------------------------------
## Method `MatCon$ecnu`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ecnu(2)

## ------------------------------------------------
## Method `MatCon$ecnp`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$ecnp(2)
```

```
## ------------------------------------------------
## Method `MatCon$aau`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$aau()

## ------------------------------------------------
## Method `MatCon$Normalize`
## ------------------------------------------------

A<-matrix(c(36,1,0,0,2,0,0,1,20),nrow=3,ncol=3)
p<-MatCon$new(A)
p$Normalize(100)
```

Index