Package ‘VUROCS’

April 7, 2020

Type Package
Title Volume under the ROC Surface for Multi-Class ROC Analysis
Version 1.0
Date 2020-04-03
Description Calculates the volume under the ROC surface and its (co)variance for ordered multi-class ROC analysis as well as certain bivariate ordinal measures of association.
License GPL-3
Imports Rcpp, doParallel, foreach
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 7.0.2
NeedsCompilation yes
Author Hannes Kazianka [cre, aut],
Anna Morgenbesser [aut],
Thomas Nowak [aut]
Maintainer Hannes Kazianka <hkazianka@gmail.com>
Repository CRAN
Date/Publication 2020-04-07 11:50:06 UTC

R topics documented:

VUROCS-package ......................................................... 2
clar ................................................................. 3
clarAdj .......................................................... 4
Kendall_taub ....................................................... 4
Kendall_tauc ...................................................... 5
Kruskal_Gamma .................................................... 6
SomersD .......................................................... 7
VUS .............................................................. 8
VUScov .......................................................... 9
VUSvar .......................................................... 10

Index 11
Description

Calculates the volume under the ROC surface and its (co)variance for ordered multi-class ROC analysis as well as certain bivariate ordinal measures of association.

Details

The package VUROCS provides three core functions to determine the volume under the ROC surface (VUS) as well as the variance and covariance of the VUS. The implementation is generally based on the algorithms presented in Waegeman, De Baets and Boullart (2008).

- \( VUS(y, fx) \) calculates the VUS for a vector of realizations \( y \) and a vector of predictions \( fx \).
- \( VUS\text{var}(y, fx) \) calculates the variance of VUS for a vector of realizations \( y \) and a vector of predictions \( fx \).
- \( VUS\text{cov}(y, fx_1, fx_2) \) calculates the covariance of the two VUS implied by the predictions \( fx_1 \) and \( fx_2 \) for a vector of realizations \( y \).

In addition to these three core functions, the package also provides an implementation of the cumulative LGD accuracy ratio (CLAR) suggested by Ozdemir and Miu (2009) specially for the purpose of assessing the discriminatory power of Loss Given Default (LGD) credit risk models. The CLAR as well as an adjusted version are computed by the functions \texttt{clar} and \texttt{clarAdj}. Moreover, the package provides time-efficient implementations of Somers’ D, Kruskall’s Gamma, Kendall’s Tau\(_b\) and Kendall’s Tau\(_c\) in the functions \texttt{SomersD}, \texttt{Kruskal_Gamma}, \texttt{Kendall\_taub} and \texttt{Kendall\_tauc}. These functions also compute asymptotic standard errors defined by Brown and Benedetti (1977) and Goktas and Oznur (2011).

Author(s)

Kazianka Hannes, Morgenbesser Anna, Nowak Thomas

References


Examples

```r
y <- rep(1:5, each=3)
fx <- c(3,3,3,rep(2:5,each=3))

VUS(y, fx)
clar(y, fx)
clarAdj(y, fx)
SomersD(y, fx)
Kruskal_Gamma(y, fx)
Kendall_taub(y, fx)
Kendall_tauc(y, fx)

VUSvar(rep(1:5, each=3), c(1,2,3,rep(2:5,each=3)))
VUScov(c(1,2,1,3,2,3), c(1,2,3,4,5,6), c(1,3,2,4,6,5))
```

---

clar  

**Cumulative LGD Accuracy Ratio**

---

**Description**

Calculates for a vector of realized categories `y` and a vector of predicted categories `hx` the cumulative LGD accuracy ratio (CLAR) according to Ozdemir and Miu 2009.

**Usage**

```
clar(y, hx)
```

**Arguments**

- `y`  
  A vector of realized values.

- `hx`  
  A vector of predicted values.

**Value**

The function returns the CLAR for a vector of realized categories `y` and a vector of predicted categories `hx`.

**References**


**Examples**

```
clar(rep(1:5, each=3), c(3,3,3,rep(2:5,each=3)))
```
**clarAdj**  
*Adjusted Cumulative LGD Accuracy Ratio*

**Description**
Calculates for a vector of realized categories \( y \) and a vector of predicted categories \( h_x \) the cumulative LGD accuracy ratio (CLAR) according to Ozdemir and Miu (2009) and adjusts it such that the measure has a value of zero if the two ordinal rankings are in reverse order.

**Usage**
```
clarAdj(y, hx)
```

**Arguments**
- **y**: a vector of realized categories.
- **hx**: a vector of predicted categories.

**Value**
The function returns the adjusted CLAR for a vector of realized categories \( y \) and a vector of predicted categories \( h_x \).

**References**

**Examples**
```
clarAdj(rep(1:5,each=3),c(3,3,3,rep(2:5,each=3)))
```

---

**Kendall_taub**  
*Kendall’s \( \tau_b \) and its asymptotic standard errors*

**Description**
Computes Kendall’s \( \tau_b \) on a given cartesian product \( Y \times f(X) \), where \( Y \) consists of the components of \( y \) and \( f(X) \) consists of the components of \( f_x \). Furthermore, the asymptotic standard error as well as the modified asymptotic standard error to test the null hypothesis that the measure is zero are provided as defined in Brown and Benedetti (1977).

**Usage**
```
Kendall_taub(y, fx)
```
Arguments

\begin{itemize}
  \item \texttt{y} a vector of realized categories.
  \item \texttt{fx} a vector of predicted values of the ranking function \texttt{f}.
\end{itemize}

Value

A list of length three is returned, containing the following components:

\begin{itemize}
  \item \texttt{val} Kendall’s Tau_b
  \item \texttt{ASE} the asymptotic standard error of Kendall’s Tau_b
  \item \texttt{ASE0} the modified asymptotic error of Kendall’s Tau_b under the null hypothesis
\end{itemize}

References


Examples

\begin{verbatim}
Kendall_taub(rep(1:5,each=3),c(3,3,3,rep(2:5,each=3)))
\end{verbatim}

\begin{verbatim}
Kendall_tauc
Kendall’s Tau_c and its asymptotic standard errors
\end{verbatim}

Description

Computes Kendall’s Tau_c on a given cartesian product \( Y \times f(X) \), where \( Y \) consists of the components of \texttt{y} and \( f(X) \) consists of the components of \texttt{fx}. Furthermore, the asymptotic standard error as well as the modified asymptotic standard error to test the null hypothesis that the measure is zero are provided as defined in Brown and Benedetti (1977).

Usage

\begin{verbatim}
Kendall_tauc(y, fx)
\end{verbatim}

Arguments

\begin{itemize}
  \item \texttt{y} a vector of realized categories.
  \item \texttt{fx} a vector of predicted values of the ranking function \texttt{f}.
\end{itemize}

Value

A list of length three is returned, containing the following components:

\begin{itemize}
  \item \texttt{val} Kendall’s Tau_c
  \item \texttt{ASE} the asymptotic standard error of Kendall’s Tau_c
  \item \texttt{ASE0} the modified asymptotic error of Kendall’s Tau_c under the null hypothesis
References


Examples

Kendall_tauc(rep(1:5,each=3),c(3,3,3,rep(2:5,each=3)))

Kruskal_Gamma

Kruskal’s Gamma and its asymptotic standard errors

Description

Computes Kruskal’s Gamma on a given cartesian product Y x f(X), where Y consists of the components of y and f(X) consists of the components of fx. Furthermore, the asymptotic standard error as well as the modified asymptotic standard error to test the null hypothesis that the measure is zero are provided as defined in Brown and Benedetti (1977).

Usage

Kruskal_Gamma(y, fx)

Arguments

y

a vector of realized categories.

fx

a vector of predicted values of the ranking function f.

Value

A list of length three is returned, containing the following components:

val

Kruskal’s Gamma

ASE

the asymptotic standard error of Kruskal’s Gamma

ASE0

the modified asymptotic error of Kruskal’s Gamma under the null hypothesis

References


Examples

Kruskal_Gamma(rep(1:5,each=3),c(3,3,3,rep(2:5,each=3)))
**SomersD**

**Somers’ D and its asymptotic standard errors**

**Description**

Computes Somers’ D on a given cartesian product Y x f(X), where Y consists of the components of \( y \) and f(X) consists of the components of \( f(x) \). Furthermore, the asymptotic standard error as well as the modified asymptotic standard error to test the null hypothesis that the measure is zero are provided as defined in Goktas and Oznur (2011).

**Usage**

SomersD(y, fx)

**Arguments**

- **y** a vector of realized categories.
- **fx** a vector of predicted values of the ranking function f.

**Value**

A list of length three is returned, containing the following components:

- **val** Somers’ D
- **ASE** the asymptotic standard error of Somers’ D
- **ASE0** the modified asymptotic error of Somers’ D under the null hypothesis.

**References**


**Examples**

SomersD(rep(1:5,each=3),c(3,3,3,rep(2:5,each=3)))
**VUS**  
*Volume under the ROC surface*

**Description**

This function computes the volume under the ROC surface (VUS) for a vector of realisations $y$ (i.e. realised categories) and a vector of predictions $fx$ (i.e. values of the ranking function $f$) for the purpose of assessing the discriminatory power in a multi-class classification problem. This is achieved by counting the number of $r$-tuples that are correctly ranked by the ranking function $f$. Thereby, $r$ is the number of classes of the response variable $y$.

**Usage**

```r
VUS(y, fx)
```

**Arguments**

- `y` a vector of realized categories.
- `fx` a vector of predicted values of the ranking function $f$.

**Value**

The implemented algorithm is based on Waegeman, De Baets and Boullart (2008). A list of length two is returned, containing the following components:

- `val` volume under the ROC surface
- `count` counts the number of observations falling into each category

**References**


**Examples**

```r
VUS(rep(1:5,each=3),c(3,3,rep(2:5,each=3)))
```
Description

Computes the covariance of the two volumes under the ROC surface (VUS) implied by two predictions $f_1$ and $f_2$ (i.e. values of two ranking functions $f_1$ and $f_2$) for a vector of realisations $y$ (i.e. realised categories) in a multi-class classification problem.

Usage

```r
VUScov(y, fx1, fx2, ncores = 1, clusterType = "SOCK")
```

Arguments

- `y`: a vector of realized categories.
- `fx1`: a vector of predicted values of the ranking function $f_1$.
- `fx2`: a vector of predicted values of the ranking function $f_2$.
- `ncores`: number of cores to be used for parallelized computations. Its default value is 1.
- `clusterType`: type of cluster to be initialized in case more than one core is used for calculations. Its default value is "SOCK". For details regarding the different types to be used, see `makeCluster`.

Value

The implemented algorithm is based on Waegeman, De Baets and Boullart (2008). A list of length three is returned, containing the following components:

- `cov`: covariance of the two volumes under the ROC surface implied by $f_1$ and $f_2$
- `val_f1`: volume under the ROC surface implied by $f_1$
- `val_f2`: volume under the ROC surface implied by $f_2$

References


Examples

```r
VUScov(c(1,2,1,3,2,3),c(1,2,3,4,5,6),c(1,3,2,4,6,5))
```
VUSvar | Variance of the volume under the ROC surface

Description
Computes the volume under the ROC surface (VUS) and its variance for a vector of realisations $y$ (i.e. realised categories) and a vector of predictions $fx$ (i.e. values of the a ranking function $f$) for the purpose of assessing the discriminatory power in a multi-class classification problem.

Usage
VUSvar(y, fx, ncores = 1, clusterType = "SOCK")

Arguments
- **y**: a vector of realized categories.
- **fx**: a vector of predicted values of the ranking function $f$.
- **ncores**: number of cores to be used for parallelized computations. The default value is 1.
- **clusterType**: type of cluster to be initialized in case more than one core is used for calculations. The default values is "SOCK". For details regarding the different types to be used, see makeCluster.

Value
The implemented algorithm is based on Waegeman, De Baets and Boullart (2008). A list of length two is returned, containing the following components:
- **var**: variance of the volume under the ROC surface
- **val**: volume under the ROC surface

References

Examples
VUSvar(rep(1:5,each=3),c(1,2,3,rep(2:5,each=3)))
Index

*Topic package
  VUROCS-package, 2

clar, 2, 3
clarAdj, 2, 4
Kendall_taub, 2, 4
Kendall_tauc, 2, 5
Kruskal_Gamma, 2, 6
makeCluster, 9, 10
SomersD, 2, 7
VUROCS (VUROCS-package), 2
VUROCS-package, 2
VUS, 2, 8
VUScov, 2, 9
VUSvar, 2, 10