Package: trinROC (via r-universe)

August 24, 2024

Title Statistical Tests for Assessing Trinormal ROC Data

Version 0.6

Date 2022-10-20

Description Several statistical test functions as well as a function for exploratory data analysis to investigate classifiers allocating individuals to one of three disjoint and ordered classes. In a single classifier assessment the discriminatory power is compared to classification by chance. In a comparison of two classifiers the null hypothesis corresponds to equal discriminatory power of the two classifiers.

Depends R (>= 3.3.0)

Imports ggplot2, rgl, gridExtra

License LGPL-2.1

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Suggests testthat, knitr, rmarkdown, MASS, reshape

VignetteBuilder knitr

URL `https://math.uzh.ch/pages/trinROC/`

BugReports `https://git.math.uzh.ch/reinhard.furrer/trinROC/-/issues`

NeedsCompilation no

Author Samuel Noll [aut], Reinhard Furrer [aut], Benjamin Reiser [ctb], Christos T. Nakas [ctb], Annina Cincera [cre, aut]

Maintainer Annina Cincera <`annina.cincera@math.uzh.ch`>

Repository CRAN

Date/Publication 2022-10-27 23:40:02 UTC

Contents

`trinROC-package` *trinROC: Statistical Tests for Assessing Trinormal ROC Data*

Description

Several statistical test functions as well as a function for exploratory data analysis to investigate classifiers allocating individuals to one of three disjoint and ordered classes. In a single classifier assessment the discriminatory power is compared to classification by chance. In a comparison of two classifiers the null hypothesis corresponds to equal discriminatory power of the two classifiers.

Details

See `vignette("Overview", package = "trinROC")` for an overview of the package. Further, `sd()`, `var()` and `cov()` are chosen with `options(trinROC.MLE = TRUE)` according to the maximum likelihood estimates (`default`). Change to sample estimates by setting `options(trinROC.MLE = FALSE)`

Author(s)

Maintainer: Annina Cincera <`annina.cincera@math.uzh.ch`>

Authors:

- Samuel Noll <`uncle.sam@gmx.net`>
- Reinhard Furrer <`reinhard.furrer@math.uzh.ch`>

Other contributors:

- Benjamin Reiser [contributor]
- Christos T. Nakas <`cnakas@uth.gr`> [contributor]

References

Noll, S., Furrer, R., Reiser, B. and Nakas, C. T. (2019). Inference in ROC surface analysis via a trinormal model-based testing approach. *Stat*, **8**(1), e249.

See Also

Useful links:

- https://math.uzh.ch/pages/trinROC/
- Report bugs at https://git.math.uzh.ch/reinhard.furrer/trinROC/-/issues

boot.test *Bootstrap test for three-class ROC data*

Description

A statistical test function to assess three-class ROC data. It can be used for assessment of a single classifier or comparison of two independent / correlated classifiers, using the Bootstrap test.

Usage

```
boot.test(
  x1,
  y1,
  z1,
  x2 = 0,
  y2 = 0,
  z2 = 0,
  dat = NULL,
  paired = FALSE,
  n.boot = 1000,
  conf.level = 0.95,
  alternative = c("two.sided", "less", "greater")
)
```

Arguments

`x1, y1, z1` Non-empty numeric vectors of data from the healthy, intermediate and diseased class from Classifier 1.

`x2, y2, z2` Numeric vectors of data from the healthy, intermediate and diseased class from Classifier 2, only needed in a comparison of two classifiers.

`dat` A data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status. The second column contains all measurements obtained from Classifier 1 (in the case of single marker assessment). In the case of comparison of two markers, column three contains the measurementss from the Classifier.

`paired` A logical indicating whether data arose from a paired setting. If `TRUE`, each class must have equal sample size for both classifiers.

`n.boot` An integer incicating the number of bootstrap replicates sampled to obtain the variance of the VUS. Default is 1000.

`conf.level` confidence level of the interval. A numeric value between (0,1) yielding the significance level $\alpha = 1 - \texttt{conf.level}$.

`alternative` character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter. For two sided test, notice $H0 : Z = (VUS_1 - VUS_2)/(Var(VUS_1) + Var(VUS_2) - 2Cov(VUS_1, VUS_2))^{0.5}$.

Details

Based on the reference standard, the Bootstrap test assesses the discriminatory power of classifiers by comparing the volumes under the ROC surfaces (VUS). It distinguishes between single classifier assessment, where a classifier is compared to the chance plane with VUS=1/6, and comparison between two classifiers. The latter case tests the equality between VUS_1 and VUS_2. The data can arise in a unpaired or paired setting. If `paired` is `TRUE`, a correlation is introduced which has to be taken into account. Therefore the sets of the two classifiers have to have classwise equal size. The data can be input as the data frame `dat` or as single vectors `x1, y1, z1, ...`. The implemented methods to evaluate the `VUS` and `var(VUS), cov(vus.1,vus.2)` are based on the empirical model assumptions and resampling techniques. This means, there are no underlying distributions assumed in any of the classes.

Value

A list of class `"htest"` containing the following components:

`statistic` the value of the Z-statistic.

`p.value` the p-value for the test.

`estimate` a data frame containing the estimated parameters from Classifier 1 and Classifier 2 (if specified).

`null.value` a character expressing the null hypothesis.

`alternative` a character string describing the alternative hypothesis.

`method` a character string indicating what type of extended Metz–Kronman test was performed.

`data.name` a character string giving the names of the data.

`Summary` A data frame representing the number of NA's as well as the means and the standard deviations per class.

`Sigma` The covariance matrix of the VUS.

References

Nakas, C. T. and C. T. Yiannoutsos (2004). Ordered multiple-class ROC analysis with continuous measurements. *Statistics in Medicine*, **23**(22), 3437–3449.

See Also

trinROC.test, trinVUS.test.

Examples

```
data(cancer)
data(krebs)

# investigate a single marker:
boot.test(dat = krebs[,c(1,2)], n.boot=500)

# result is equal to:
x1 <- with(krebs, krebs[trueClass==”healthy”, 2])
y1 <- with(krebs, krebs[trueClass==”intermediate”, 2])
z1 <- with(krebs, krebs[trueClass==”diseased”, 2])
boot.test(x1, y1, z1, n.boot=500)

# comparison of marker 2 and 6:
boot.test(dat = krebs[,c(1,2,5)], paired = TRUE)

# result is equal to:
x2 <- with(krebs, krebs[trueClass==”healthy”, 5])
y2 <- with(krebs, krebs[trueClass==”intermediate”, 5])
z2 <- with(krebs, krebs[trueClass==”diseased”, 5])
boot.test(x1, y1, z1, x2, y2, z2, paired = TRUE)
```

boxcoxROC *Box-Cox transformation on three-class ROC data*

Description

A transformation function for three-class ROC data in order to obtain normally distributed classes.

Usage

```
boxcoxROC(
  x,
  y,
  z,
  lambda = seq(-2, 2, 0.05),
  lambda2 = NULL,
  eps = 0.02,
  verbose = TRUE
)
```

Arguments

`x, y, z`	vectors containing the data of the three classes "healthy", "intermediate" and "diseased" to be transformed. In two-class ROC analysis only.
`lambda`	vector of possible lambdas the log-likelihood function is evaluated.
`lambda2`	numeric shifting parameter. For the implemented Box-Cox transformation positive measurements in `x, y, z` are required. `lambda2` is used to shift these measurements.
`eps`	numeric; indicating the bandwith around zero, where `lambda` is treated to be zero and the data is log-transformed.
`verbose`	logical; indicating whether output should be displayed (default) or not.

Details

A Box-Cox transformation computing

$$X^{(\lambda)} = \begin{cases} (X^{\lambda} - 1)/\lambda, & \text{if } \lambda \neq 0, \\ \log(X), & \text{else } \lambda = 0, \end{cases}$$

with optimal λ estimated from the likelihood kernel function, as formally described in the supplementary material in Bantis et al. (2017). If the data include any nonpositive observations, a shifting parameter `lambda2` can be included in the transformation given by:

$$X^{(\lambda)} = \begin{cases} ((X + \lambda_2)^{\lambda} - 1)/\lambda, & \text{if } \lambda \neq 0, \\ \log(X + \lambda_2), & \text{else } \lambda = 0. \end{cases}$$

Value

A list with components:

`xbc, ybc, zbc`	The transformed vectors.
`lambda`	estimated optimal parameter.
`shapiro.p.value`	p-values obtained from `shapiro.test()` of the original and transformed data.

References

Bantis LE, Nakas CT, Reiser B, Myall D and Dalrymple-Alford JC (2015) Construction of joint confidence regions for the optimal true class fractions of receiver operating characteristic (roc) surfaces and manifolds. *Statistical Methods in Medical Research* **26**(3): 1429–1442.

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations (with discussion). *Journal of the Royal Statistical Society, Series B*, **26**, 211–252.

See Also

`shapiro.test` and `boxcox` from the package `MASS`.

Examples

```
data(cancer)
x1 <- with(cancer, cancer[trueClass=="healthy", 9])
y1 <- with(cancer, cancer[trueClass=="intermediate", 9])
z1 <- with(cancer, cancer[trueClass=="diseased", 9])

boxcoxROC(x1, y1, z1)
```

cancer *Synthetic data set to investigate three-class ROC data.*

Description

A dataset containing randomly generated measurements from three diagnostic classes as they may arise in a cancer investigation.

Usage

```
cancer
```

Format

A data frame with 100 rows and 10 variables (9 classifiers):

trueClass A factor, indicating the class membership of the individuals.

Class1, Class2, Class3, Class4, Class5, Class6, Class7, Class8, Class9 Measurements of a specific classifier obtained from the patients that underwent the clinical study.

emp.vus *Empirical VUS calculation*

Description

This function computes the empirical Volume Under the Surface (VUS) of three-class ROC data.

Usage

```
emp.vus(x, y, z, dat = NULL, old.version = TRUE)
```

Arguments

`x, y, z` Numeric vectors contaning the measurements from the healthy, intermediate and diseased class.

`dat` A data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status. The second column contains all measurements obtained from Classifier.

`old.version` A logical to switch computation method to the old version, which is up to 50% faster in computation (at N=50).

Details

This function computes the empirical VUS of three-class ROC data using the `expand.grid` function. It has been shown to be faster than computation using the `merge` function (`VUS.merge()`) or direct geometrical imlementation. The measurements can be input as seperate vectors `x, y, z` or as a data frame `dat`.

Value

It returns the numeric `VUS` of the data.

References

Scurfield, B. K. (1996). Multiple-event forced-choice tasks in the theory of signal detectability. *Journal of Mathematical Psychology* **40.3**, 253–269.

Nakas CT and Yiannoutsos CT (2004) Ordered multiple-class roc analysis with continuous measurements. *Statistics in Medicine* **23**(22): 3437–3449.

Examples

```
data(krebs)
x1 <- with(krebs, cancer[trueClass=="healthy", 4])
y1 <- with(krebs, cancer[trueClass=="intermediate", 4])
z1 <- with(krebs, cancer[trueClass=="diseased", 4])

emp.vus(x1, y1, z1)
# Alternatively:
emp.vus(dat = krebs[,c(1,4)])
```

findmu *Determine equidistant means of trinormal ROC data simulation*

Description

A function that computes the equidistant means `muy` and `muz` for a specific `mux`. The VUS as well as the set of standard errors are given as arguments to the function.

Usage

```
findmu(mux = 0, sdx = 1, sdy = 1, sdz = 1, VUS = 1/6, step = 0.001)
```

Arguments

mux	The numeric mean of the healthy class. Default is zero.
sdx, sdy, sdz	The numeric standard errors of the healthy, intermediate and diseased class, for which the according means have to be determined given a specifiv VUS.
VUS	The Volume Under the Surface. A numeric value between 1/6 and 1. Default is 1/6.
step	A numeric indicating the step size each iteration takes in order to find the closest set of means. Default set to 0.001.

Details

Defaults are: VUS = 1/6, standard errors for all three classes equal 1. The searching algorithm is stepwise increasing the differences muy-mux and muz-mux according to the variable step. The algorithm stops when the computed VUS exceeds the preferred VUS. The according parameters mux, muy, muz are returned with the computed VUS.

Remark: The bigger VUS and the smaller step is chosen, the longer the computation lasts.

Value

A data frame with the following components:

mux	The initial mean of the healthy class
muy	The mean of the intermediate class computed for the specified VUS.
muz	The mean of the diseased class computed for the specified VUS.
VUS	The VUS computed for mux, muy and muz.

Examples

```
# find equidistant means with mux=2.7 and VUS = 0.45:
findmu(mux = 2.7, VUS = 0.45)
# specify standard errors:
findmu(mux = 2.7, sdx = 1.1, sdy = 1.3, sdz = 1.5, VUS = 0.45)
```

---

krebs *Synthetic small data set to investigate three-class ROC data.*

Description

A dataset containing randomly generated measurements from three diagnostic classes as they may arise in a cancer investigation. For illustration, this dataset has been chosen to be smaller than the data set cancer.

Usage

```
krebs
```

Format

A data frame with 50 rows and 5 variables (4 classifiers):

trueClass A factor, indicating the class membership of the individuals.

Fac1, Fac2, Fac3, Fac4 Measurements obtained from the patients that underwent the clinical study.

`roc.eda` *Exploratory data analysis for a three-class ROC marker*

Description

A function that investigates data that arose from a single marker and containes the reference standard of the three classes "healthy", "intermediate" and "diseased".

Usage

```
roc.eda(
  x,
  y,
  z,
  dat = NULL,
  type = c("empirical", "trinormal"),
  plotVUS = FALSE,
  saveVUS = FALSE,
  sep.dens = FALSE,
  scatter = FALSE,
  conf.level = 0.95,
  n.boot = 1000,
  verbose = TRUE,
  alternative = c("two.sided", "less", "greater")
)
```

Arguments

`x, y, z` numeric vectors contaning the measurements from the healthy, intermediate and diseased class.

`dat` a data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status.

`type` a character, specifying if the `empirical` VUS and tests or the `trinormal` VUS and tests are computed.

`plotVUS` a logical whether to evaluate and plot the VUS (default is `FALSE`). Note: To save a png `plotVUS` needs to be `TRUE` too.

`saveVUS` a logical whether to save a PNG of the VUS in your current working directory (default is `FALSE`).

`sep.dens` a logical indicating if the densitie plots should be plotted on separate x-axes (`TRUE`) or on a common axe (`FALSE`, is default).

`scatter` a logical indicating if the measurements per class plot should be plotted as a boxplot (default) or as a scatterplot (`scatter = TRUE`).

`conf.level` A numeric value between 0 and 1 yielding the significance level $\alpha = 1 -$ `conf.level`.

`n.boot` an integer incicating the number of bootstrap replicates sampled to obtain the variance of the VUS. Default is 1000.

`verbose` a logical, indicating whether output should be displayed or not. Default is `TRUE`.

`alternative` a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`.

Details

For the preliminary assessment of a classifier, exporatory data analysis (EDA) on the markers is necessary. This function assesses measurements from a single marker and computes the VUS, statistical tests and returns a summary table as well as some plots of the data.

Value

A list with class "htest" containing the following components:

`statistic` The value of the test(s).

`p.value` The p-value for the test(s).

`VUS` the VUS computed with the specific method defined in `type`.

`dat.summary` A data frame displaying size, mean and standard deviation of the three classes.

`alternative` The alternative hypothesis.

`type` a character containing the the method used for the exploratory data analysis.

`data.name` a character containing the name of the data.

`xVUS, yVUS, zVUS` (if `plotVUS = TRUE`) numeric vectors and matrices computed by `rocsurf.emp` or `rocsurf.trin`, used for displaying the surface with package `rgl`.

`histROC` a `ggplot2` object, displaying the historgrams and densities of the three classes.

`meas.overview` A ggplot2 object, displaying the boxplots (if `scatter = FALSE`) or scatter plots of the three classes (if `scatter = TRUE`).

Warning

If `type = "empirical"`, computation may take a while, as `roc.eda` calls the function `boot.test()`.

See Also

trinROC.test, trinVUS.test for trinormal data investigation, boot.test for empirical data analysis. rocsurf.emp, rocsurf.trin for the surface plot.

Examples

```
data(krebs)

# empirical EDA:
roc.eda(dat = krebs[,c(1,5)], type = "e", plotVUS = FALSE)

# equal data input via:
x <- with(krebs, krebs[trueClass=="healthy", 5])
y <- with(krebs, krebs[trueClass=="intermediate", 5])
z <- with(krebs, krebs[trueClass=="diseased", 5])
roc.eda(x, y, z, type = "e", sep.dens = TRUE)

data(cancer)
# trinormal EDA:
roc.eda(dat = cancer[,c(1,10)], type = "trin", plotVUS = FALSE)
# trinormal EDA with different plots:
roc.eda(dat = cancer[,c(1,5)], type = "t", sep.dens = TRUE, scatter = TRUE)
```

roc3.test

Statistical test function for computing multiple tests on three-class ROC data

Description

A statistical test function that assesses three-class ROC data with the trinormal based ROC test, the trinormal VUS test and the Bootstrap test.

Usage

```
roc3.test(
  dat,
  type = c("ROC", "VUS", "Bootstrap"),
  paired = FALSE,
  conf.level = 0.95,
  n.boot = 1000,
  p.adjust = FALSE
)
```

Arguments

dat A data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status.

`type` A character, specifying which tests are applied to `dat`. `"ROC"` implies the trinormal based ROC test, `"VUS"` the trinormal VUS test and `"Bootstrap"` the Bootstrap test.

`paired` A logical indicating whether data arose from a paired setting. If data is paired, each class must have equal sample size for both classifiers.

`conf.level` confidence level of the interval. A numeric value between (0,1) yielding the significance level $\alpha = 1 - \texttt{conf.level}$.

`n.boot` An integer incicating the number of Bootstrap replicates sampled to obtain the variance of the VUS. Default is 1000.

`p.adjust` A logical, indicating whether a FDR adjustment should be applied to the p-values. Default is `FALSE`.

Details

For the preliminary assessment of a classifier, different statistical tests have been proposed in the literature. This function can be used for either comparison of single classifiers to a null hypothesis of being not better than a random allocation function or comparison of two classifiers under the null hypothesis of having equal discriminatory power. Depending on the specification of the user, (s)he can apply the trinormal based ROC test (LINK), the test developed by Xiong et. al. or the Bootstrap test or any combination of these tests. More information of the specific tests can be obtained by calling `?functionname`. If more than two markers are present, a pairwise comparison between each marker is realized.

Value

A list with components:

`Overview` a data frame with number of columns according to number of markers. Rows contain the following information about the makers:

1. Index according to smallest VUS
2. VUS
3. P-values of statistical test specified by `type`
4. Number of NA's

`O.orig` the unsorted `Overview`.

`P.values` a list, containing the upper triangular matrices of the optionally adjusted p-values of the statistical tests chosen by `type`.

`Test.Values` a list, containing the upper triangular matrices of the test values of the statistical tests chosen by `type`.

Note

If `type = "Bootstrap"`, the Bootstrap test is evaluated. This may take some time, especially with sample sizes > 100.

Examples

```
data(krebs)
roc3.test(krebs, type = c("ROC", "VUS"), paired = TRUE)[c("Overview","P.values")]
```

rocsurf.emp *Empirical ROC surface plot*

Description

Function for computation of the empirical ROC surface.

Usage

```
rocsurf.emp(x, y, z, plot = TRUE, saveVUS = FALSE)
```

Arguments

x, y, z	Vectors containing the data of the three classes "healthy", "intermediate" and "diseased".
plot	logical. If TRUE (default), the VUS is plotted using surface3d from the package rgl.
saveVUS	A logical whether to save a PNG of the VUS in your current working directory (default is FALSE).

Details

This function takes three-class ROC data and computes the three dimentional surface using the R-package rgl. The ROC surface is defined as

$$z = ROCs(t_-, t_+) = F_0(c_+) - F_0(c_-) = F_0(G_+^{-1}(t_+)) - F_0(F_-^{-1}(t_-)),$$

where c_-, c_+ are the two cut-off points and F_-, F_0, F_+ the cdf of the three classes with $G = 1 - F$.

Value

A list with the following components:

t1, t2, zVUS	The matice containing the surface values.
x, y, z	The original data.

See Also

surface3d.

Examples

```
data(cancer)
x1 <- with(cancer, cancer[trueClass=="healthy", 9])
y1 <- with(cancer, cancer[trueClass=="intermediate", 9])
z1 <- with(cancer, cancer[trueClass=="diseased", 9])

rocsurf.emp(x1, y1, z1)
```

`rocsurf.trin` *Trinormal ROC surface plot*

Description

Function for computation of the trinormal ROC surface.

Usage

```
rocsurf.trin(x, y, z, p = 300, plot = TRUE, saveVUS = FALSE)
```

Arguments

`x, y, z`	Vectors containing the data of the three classes "healthy", "intermediate" and "diseased".
`p`	An integer for the precision of the surface. `p` gives the number of gridpoints per axis.
`plot`	logical. If TRUE (default), the VUS is plotted using `surface3d` from the package `rgl`.
`saveVUS`	A logical whether to save a PNG of the VUS in your current working directory (default is `FALSE`).

Details

This function takes three-class ROC data and computes the three dimentional surface using the R-package `rgl`. The ROC surface is defined as

$$z = ROCs(t_-, t_+) = F_0(c_+) - F_0(c_-) = F_0(G_+^{-1}(t_+)) - F_0(F_-^{-1}(t_-)),$$

where c_-, c_+ are the two cut-off points and F_-, F_0, F_+ the cdf of the three classes with $G = 1 - F$.

Value

A list with the following components:

`t1, t2`	The vectors $t_- = F_-^{(c_-)}$ and $t_+ = F_+^{(c_+)}$
`zVUS`	The matix containing the surface values.
`x, y, z`	The original data.

References

Xiong, C., G. Van Belle, et al. (2006). Measuring and estimating diagnostic accuracy when there are three ordinal diagnostic groups. *Statistics in Medicine* 25(7), 1251–1273.

Examples

```
data(cancer)
x1 <- with(cancer, cancer[trueClass=="healthy", 8])
y1 <- with(cancer, cancer[trueClass=="intermediate", 8])
z1 <- with(cancer, cancer[trueClass=="diseased", 8])

rocsurf.trin(x1, y1, z1)
```

`trinROC.test` *Trinormal based ROC test*

Description

A statistical test function to assess three-class ROC data. It is possible to investigate a single classifier or make a comparison of two independent / correlated classifiers.

Usage

```
trinROC.test(
  x1,
  y1,
  z1,
  x2 = 0,
  y2 = 0,
  z2 = 0,
  dat = NULL,
  paired = FALSE,
  conf.level = 0.95
)
```

Arguments

`x1, y1, z1` (non-empty) numeric vectors of data from the healthy, intermediate and diseased class from Classifier 1.

`x2, y2, z2` numeric vectors of data from the healthy, intermediate and diseased class from Classifier 2.

`dat` a data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status. The second column contains all measurements obtained from Classifier 1 (in the case of single marker assessment). In the case of comparison of two markers, column three contains the measurementss from the Classifier.

`paired` a logical indicating whether data arose from a paired setting. If `TRUE`, each class must have equal sample size for both classifiers.

`conf.level` confidence level of the interval. A numeric value between (0,1) yielding the significance level $\alpha = 1-$`conf.level`.

Details

The trinormal ROC model is a parametric model in three-class ROC analysis. It is based on normality in each of the trhee classes D_- (healthy), D_0 (intermediate) and D_+ (diseased) with denoted distributions $N(\mu_-, \sigma_-^2)$, $N(\mu_0, \sigma_0^2)$ and $N(\mu_+, \sigma_+^2)$. A classifier of a trinormal ROC model classifies individuals into one of the three ordered classes based on two cut-off points $c_- < c_+$. We define $t_- = F_-(c_-)$ and $t_+ = 1 - F_+(c_+) = G_+(c_+)$. Now, the ROC surface can be written as

$$ROCs(t_-, t_+) = \Phi\left(\frac{\Phi^{-1}(1 - t_+) + d}{c}\right) - \Phi\left(\frac{\Phi^{-1}(t_-) + b}{a}\right)$$

whith parameters a, b, c and c given by $a = \frac{\hat{\sigma}_0}{\hat{\sigma}_-}, b = \frac{\hat{\mu}_- - \hat{\mu}_0}{\hat{\sigma}_-}, c = \frac{\hat{\sigma}_0}{\hat{\sigma}_+}, d = \frac{\hat{\mu}_+ - \hat{\mu}_0}{\hat{\sigma}_+}$. It is a surface in the unit cube that plots the probability of a measurement to get assigned to the intermediate class as the two thresholds c_-, c_+ are varying.

Based on the reference standard, the trinormal based ROC test can be used to assess the discriminatory power of such classifiers. It distinguishes between single classifier assessment, where a classifier is compared to some hypothetical distributions in the classes, and comparison between two classifiers. The latter case tests for equality between the parameters a, b, c and d of the ROC curves. The data can arise in a unpaired or paired setting. If `paired` is `TRUE`, a correlation is introduced which has to be taken into account. Therefore the sets of the two classifiers have to have classwise equal size. The data can be input as the data frame `dat` or as single vectors `x1, y1, z1, ...`.

As the Chi-squared test is by definition a one-sided test, the variable `alternative` cannot be specified in this test. For this 'goodness of fit' test, we assume the parameters $a_1, \ldots, d_1$ and $a_2, \ldots, d_2$ to have a pairwise equivalent normal distribution (in large sample sets).

Value

A list of class `"htest"` containing the following components:

`statistic` the value of the chi-squared statistic.

`parameter` the degrees of freedom for the chi-squared statistic.

`p.value` the p-value for the test.

`conf.int` a confidence interval for the test.

`estimate` a data frame containing the estimated VUS and parameters a, b, c and d from Classifier 1 and Classifier 2 (if specified).

`null.value` a character expressing the null hypothesis.

`alternative` a character string describing the alternative hypothesis.

`method` a character string indicating what type of trinormal based ROC test was performed.

`data.name` a character string giving the names of the data.

`CovMat` the covariance matrix of the chi-squared statistic.

`Summary` a data frame representing the number of NA's as well as the means and the standard deviations per class.

References

Noll, S., Furrer, R., Reiser, B. and Nakas, C. T. (2019). Inference in ROC surface analysis via a trinormal model-based testing approach. *Stat*, **8**(1), e249.

See Also

`trinVUS.test`, `boot.test`.

Examples

```
data(cancer)
data(krebs)

# investigate a single marker:
trinROC.test(dat = cancer[,c(1,3)])
trinROC.test(dat = krebs[,c(1,5)])

# result is equal to:
x1 <- with(cancer, cancer[trueClass=="healthy", 3])
y1 <- with(cancer, cancer[trueClass=="intermediate", 3])
z1 <- with(cancer, cancer[trueClass=="diseased", 3])
trinROC.test(x1, y1, z1)

# comparison of marker 2 and 6:
trinROC.test(dat = cancer[,c(1,3,5)], paired = TRUE)
trinROC.test(dat = cancer[,c(1,3,5)], paired = FALSE)

# result is equal to:
x2 <- with(cancer, cancer[trueClass=="healthy", 5])
y2 <- with(cancer, cancer[trueClass=="intermediate", 5])
z2 <- with(cancer, cancer[trueClass=="diseased", 5])
trinROC.test(x1, y1, z1, x2, y2, z2, paired = TRUE)
```

`trinVUS.test` *Trinormal VUS test*

Description

A statistical test function to assess three-class ROC data. It can be used for assessment of a single classifier or comparison of two independent / correlated classifiers, using the statistical test developed by Xiong et al. (2007).

Usage

```
trinVUS.test(
  x1,
  y1,
  z1,
  x2 = 0,
  y2 = 0,
  z2 = 0,
  dat = NULL,
  paired = FALSE,
  conf.level = 0.95,
  alternative = c("two.sided", "less", "greater")
)
```

Arguments

x1, y1, z1 non-empty numeric vectors of data from the healthy, intermediate and diseased class from Classifier 1.

x2, y2, z2 numeric vectors of data from the healthy, intermediate and diseased class from Classifier 2, only needed in a comparison of two classifiers.

dat a data frame of the following structure: The first column represents a factor with three levels, containing the true class membership of each measurement. The levels are ordered according to the convention of higher values for more severe disease status. The second column contains all measurements obtained from Classifier 1 (in the case of single marker assessment). In the case of comparison of two markers, column three contains the measurementss from the Classifier.

paired logical; indicating whether data arose from a paired setting. If TRUE, each class must have equal sample size for both classifiers.

conf.level confidence level of the interval. A numeric value between (0,1) yielding the significance level $\alpha = 1 - \text{conf.level}$.

alternative character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter. For two sided test, notice $H0 : Z = (VUS_1 - VUS_2)/(Var(VUS_1) + Var(VUS_2) - 2Cov(VUS_1, VUS_2))^{0.5}$.

Details

Based on the reference standard, this trinormal VUS test assesses the discriminatory power of classifiers by comparing the volumes under the ROC surfaces (VUS). It distinguishes between single classifier assessment, where a classifier is compared to the chance plane with VUS=1/6, and comparison between two classifiers. The latter case tests the equality between VUS_1 and VUS_2. The data can arise in a unpaired or paired setting. If paired is TRUE, a correlation is introduced which has to be taken into account. Therefore the sets of the two classifiers have to have classwise equal size. The data can be input as the data frame dat or as single vectors x1, y1, z1, ...

Value

A list of class "htest" containing the following components:

statistic the value of the Z-statistic.

p.value the p-value for the test.

conf.int a confidence interval for the test.

`estimate` a data frame containing the estimated VUS from Classifier 1 and Classifier 2 (if specified).

`null.value` a character expressing the null hypothesis.

`alternative` a character string describing the alternative hypothesis.

`method` a character string indicating what type of trinormal VUS test was performed.

`data.name` a character string giving the names of the data.

`Summary` a data frame representing the number of NA's as well as the means and the standard deviations per class.

`Sigma` the covariance matrix of the VUS.

References

Xiong, C., Van Belle, G. Miller J. P., Morris, J. C. (2006). Measuring and estimating diagnostic accuracy when there are three ordinal diagnostic groups. *Statistics in Medicine*, **25**(7), 1251–1273.

Xiong, C., van Belle, G., Miller, J. P., Yan, Y., Gao, F., Yu, K., and Morris, J. C. (2007). A parametric comparison of diagnostic accuracy with three ordinal diagnostic groups. *Biometrical Journal*, **49**(5), 682–693. doi: 10.1002/bimj.200610359.

See Also

`trinROC.test`, `boot.test`.

Examples

```
data(cancer)
data(krebs)

# investigate a single marker:
trinVUS.test(dat = cancer[,c(1,3)])
trinVUS.test(dat = krebs[,c(1,5)])

# result is equal to:
x1 <- with(cancer, cancer[trueClass=="healthy", 3])
y1 <- with(cancer, cancer[trueClass=="intermediate", 3])
z1 <- with(cancer, cancer[trueClass=="diseased", 3])
trinVUS.test(x1, y1, z1)

# comparison of marker 2 and 6:
trinVUS.test(dat = cancer[,c(1,3,5)], paired = TRUE)
trinVUS.test(dat = cancer[,c(1,3,5)], paired = FALSE)

# result is equal to:
x2 <- with(cancer, cancer[trueClass=="healthy", 5])
y2 <- with(cancer, cancer[trueClass=="intermediate", 5])
z2 <- with(cancer, cancer[trueClass=="diseased", 5])
trinVUS.test(x1, y1, z1, x2, y2, z2, paired = TRUE)
```

Index