Package ‘comparison’

October 12, 2022

Version 1.0-5

Date 2020-08-04

Title Multivariate Likelihood Ratio Calculation and Evaluation

Encoding UTF-8

Description Functions for calculating and evaluating likelihood ratios from uni/multivariate continuous observations.
The package includes the two-level functions to calculate the LR assuming multivariate normality, and another with drops this assumption and uses a multivariate kernel density estimate. The package also contains code to perform empirical cross entropy (ECE) calibration of likelihood ratios. The LR functions are based primarily on Aitken, C.G.G. and Lucy, D. (2004) <doi:10.1046/j.0035-9254.2003.05271.x>, ``Evaluation of trace evidence in the form of multivariate data,'' Journal of the Royal Statistical Society: Series C (Applied Statistics), 53: 109-122. The ECE functions are based primarily on D. Ramos and J. Gonzalez-Rodrigues, (2008) ``Cross-entropy analysis of the information in forensic speaker recognition,'' in Proc. IEEE Odyssey, Speaker Lang. Recognit. Workshop.

Depends R (>= 3.5.0)

Imports isotone

License GPL (>= 2)

URL github.com/jmcurran/comparison

LazyLoad yes

RoxygenNote 7.1.1

NeedsCompilation no

Author David Lucy [aut],
James Curran [aut, cre],
Agnieszka Martyna [aut]

Maintainer James Curran <j.curran@auckland.ac.nz>

Repository CRAN

Date/Publication 2020-08-05 11:40:02 UTC

R topics documented:

`calc.ece` *Empirical cross-entropy (ECE) calculation*

Description

Calculates the empirical cross-entropy (ECE) for likelihood ratios from a sequence same and different item comparisons.

Usage

```
calc.ece(LR.ss, LR.ds, prior = seq(from = 0.01, to = 0.99, length = 99))
```

Arguments

`LR.ss`	a vector of likelihood ratios (LRs) from same source calculations
`LR.ds`	a vector of LRs from different source calculations
`prior`	a vector of ordinates for the prior in ascending order, and between 0 and 1. Default is 99 divisions of 0.01 to 0.99.

Details

Acknowledgements:
The function to calculate the values of the likelihood ratio for the `calibrated.set` draws heavily upon the `opt_loglr.m` function from Niko Brummer's FoCal package for Matlab.

Value

Returns an S3 object of class `ece`

Author(s)

David Lucy

References

@references D. Ramos and J. Gonzalez-Rodrigues, (2008) "Cross-entropy analysis of the information in forensic speaker recognition," in Proc. IEEE Odyssey, Speaker Lang. Recognit. Workshop. Zadora, G. & Ramos, D. (2010) Evaluation of glass samples for forensic purposes - an application of likelihood ratio model and information-theoretical approach. Chemometrics and Intelligent Laboratory: 102; 63-83.

See Also

`isotone::gpava()`, `calibrate.set()`

Examples

```
LR.same = c(0.5, 2, 4, 6, 8, 10)  # the same has 1 LR < 1
LR.different = c(0.2, 0.4, 0.6, 0.8, 1.1)  # the different has 1 LR > 1
ece.1 = calc.ece(LR.same, LR.different) # simplest invocation
plot(ece.1) # use plot method
```

`calibrate.set` *Calculate the calibrated set of idea LRs*

Description

Calculates and returns the calibrated set of `ideal' LRs from the observed LRs using the penalised adjacent violato` loglr()‘ function for Matlab.

Usage

```
calibrate.set(LR.ss, LR.ds, method = c("raw", "laplace"))
```

Arguments

`LR.ss`	a vector of likelihood ratios for the comparisons of items known to be from the same source
`LR.ds`	a vector of likelihood ratios for the comparisons of items known to be from different sources
`method`	the method used to perform the calculation, either `"raw"` or `"laplace"`

Details

This is an internal function, and is not meant to be called directly. However it has been exported just in case.

Value

a `list` with two items:

LR.cal.ss calibrated LRs for the comparison for same set

LR.cal.ds calibrated LRs for the comparison for different set

Author(s)

David Lucy

References

D. Ramos and J. Gonzalez-Rodrigues, (2008) "Cross-entropy analysis of the information in forensic speaker recognition," in Proc. IEEE Odyssey, Speaker Lang. Recognit. Workshop.

See Also

`isotone::gpava()`, `calc.ece()`

`comparison` *comparison: A package for computing likelihood ratios for univariate and multivariate evidence.*

Description

This package is for computing the *weight of the evidence*, i.e. the *likelihood ratio (LR)* for trace evidence which has been quantified with some instrument. For example a forensic scientist might be have determined the refractive indices of fragments of glass taken from a crime scene and fragments of glass recovered from the clothing of the suspected breaker. This package evaluates the probability (density) of the evidence, E, (the RI values from the two samples) under the hypothesis H_p that they originated from the same source, and alternatively under the hypothesis H_d that they originated from another source. The LR is the ratio of these two quantities, i.e.

$$LR = \frac{p(E|H_p)}{p(E|H_d)}$$

. A LR which is greater than one indicates that the evidence supports H_p, and a LR which is less than one indicates that the evidence supports H_d.

Details

The computation can use either univariate or multivariate observations of a physical object. For example trace element measurements, and a similar set of uni/multivariate observations from another object, and calculates a likelihood ratio for the propositions that the first item came from the same source as the second given some population data.

Acknowledgements:

In a package of functions such as these which have undergone a long development over a number of years, it is inevitable that a number of people, besides those directly cited, have helped to correct and add to the code. These people are (in alphabetical order): Ivo Alberink (NFI), Anabel Bolck (NFI), Sonja Menges (BKA), Geoff Morrison (Aston), Tereza Neocleous (Glasgow), Anders Nordgaard (SKL), Brad Patterson (George Mason), Phil Rose (ANU), Agnieszka Rzepecka (Jagiellonian), Marjan Sjerps (NFI) and Hanjing Zhang (Edinburgh).

References

Aitken, C.G.G. & Lucy, D. (2004) Evaluation of trace evidence in the form of multivariate data. Applied Statistics: 53(1):109-122.

`glass` *Glass composition data for seven elements from 200 glass items.*

Description

These data are from Grzegorz (Greg) Zadora at the Institute of Forensic Research in Krakow, Poland. They are the log of the ratios of each element to oxygen, so logNaO is the log(10) of the Sodium to Oxygen ratio, and logAlO is the log of the Aluminium to Oxygen ratio. The instrumental method was SEM-EDX.

Usage

```
data(glass)
```

Format

a `data.frame` with 2400 rows and 9 columns.

item factor200 levels - which item the measurements came from

fragment factor4 levels - which of the four fragments from each item the observations were made upon

logNaO numericlog of sodium concentration to oxygen concentration

logMgO numericlog of magnesium concentration to oxygen concentration

logAlO numericlog of aluminium concentration to oxygen concentration

logSiO numericlog of silicon concentration to oxygen concentration

logKO numericlog of potassium concentration to oxygen concentration

logCaO numericlog of calcium concentration to oxygen concentration

logFeO numericlog of iron concentration to oxygen concentration

Details

The `item` indicates the object the glass came from. The levels for each item are unique to that item. The `fragment` can be considered a sub-item. When collecting these observations Greg took a glass object, say a jam jar, he would then break it, and extract four fragments. Each fragment would be measured three times upon different parts of that fragment. The fragment labels are repeated, so, for example, fragment "f1" from item "s2" has nothing whatsoever to do with fragment "f1" from item "s101".

For two level models use `item` as the lower level - three level models can use the additional information from the individual fragments.

Source

Grzegorz Zadora Institute of Forensic Research, Krakow, Poland.

References

Aitken, C.G.G. Zadora, G. & Lucy, D. (2007) A Two-Level Model for Evidence Evaluation. *Journal of Forensic Sciences*: **52**(2); 412-419.

`plot.ece` *An S3 plot method for objects of class* `ece`

Description

An S3 plot method for objects of class `ece`

Usage

```
## S3 method for class 'ece'
plot(x, ...)
```

Arguments

`x`	an S3 object of class `ece` which is generated from `calc.ece()`.
`...`	other arguments that are passed to the `plot` generic.

Author(s)

David Lucy

See Also

`calc.ece()`

`two.level.comparison.items` *Create a* `compitem` *object.*

Description

This function creates a `compitem` object from a `data.frame` or `matrix` of observations from an item to be deemed a control, or a recovered, item.

Usage

```
two.level.comparison.items(data, data.columns)
```

Arguments

`data`	a `matrix` or `data.frame` of observed properties from either the control item, or the recovered item
`data.columns`	vector of integers giving which columns in `data` are the observations of the properties

Value

an object of class `compitem`

Examples

```
# load Greg Zadora's glass data
data(glass)

# calculate a compitem object representing the control item
control = two.level.comparison.items(glass[1:6,], c(7,8,9))
```

`two.level.components` *Compute integrated means and covariances*

Description

Takes a large sample from the background population and calculates the within and between covariance matrices, a vector of means, a vector of the counts of replicates for each item from the sample, and other bits needed to make up a `compcovar` object.

Usage

```
two.level.components(data, data.columns, item.column)
```

Arguments

`data`	a `matrix`, or `data.frame`, of observations, with cases in rows, and properties as columns
`data.columns`	a `vector` indicating which columns are the properties
`item.column`	an integer indicating which column gives the item

Details

Uses ML estimation at the moment - this will almost certainly change in the future and hopefully allow regularisation methods to get a more stable (and non-singular) estimate.

Value

an object of class `compvar`

Examples

```
# load Greg Zadora's glass data
data(glass)

# calculate a compcovar object based upon glas
# using K, Ca and Fe - warning - could take time
# on slower machines
Z = two.level.components(glass, c(7,8,9), 1)
```

---

two.level.density.LR *Calculate the likelihood ratio using multivariate KDEs*

Description

Takes a `compitem` object which represents some control item, and a `compitem` object which represents a recovered item, then uses information from a `compcovar` object, which represents the information from the population, to calculate a likelihood ratio as a measure of the evidence given by the observations for the same/different source propositions.

Usage

```
two.level.density.LR(control, recovered, background)
```

Arguments

`control`	a `compitem` object with the control item information
`recovered`	a `compitem` object with the recovered item information
`background`	a `compcovar` object with the population information

Value

an estimate of the likelihood ratio

References

Aitken, C.G.G. & Lucy, D. (2004) Evaluation of trace evidence in the form of multivariate data. *Applied Statistics*: **53**(1); 109-122.

Examples

```
library(comparison)
# load Greg Zadora's glass data
data(glass)

# calculate a compcovar object based upon glass
# using K, Ca and Fe - warning - could take time
# on slower machines
Z = two.level.components(glass, c(7,8,9), 1)
```

```
# calculate a compitem object representing the control item
control = two.level.comparison.items(glass[1:6,], c(7,8,9))

# calculate a compitem object representing the recovered item
# known to be from the same item (item 1)
recovered.1 = two.level.comparison.items(glass[7:12,], c(7,8,9))

# calculate a compitem object representing the recovered item
# known to be from a different item (item 2)
recovered.2 = two.level.comparison.items(glass[19:24,], c(7,8,9))


# calculate the likelihood ratio for a known
# same source comparison - should be 20.59322
# 2020-08-01 Both this version and the previous version return 20.58967
lr.1 = two.level.density.LR(control, recovered.1, Z)
lr.1

# calculate the likelihood ratio for a known
# different source comparison - should be 0.02901532
# 2020-08-01 Both this version and the previous version return 0.01161392
lr.2 = two.level.density.LR(control, recovered.2, Z)
lr.2
```

---

two.level.lindley.LR *Likelihood ratio calculation using Lindley's approach*

Description

Takes a `compitem` object which represents some control item, and a `compitem` object which represents a recovered item, then uses information from a `compcovar` object, which represents the information from the population, to calculate a likelihood ratio as a measure of the evidence given by the observations for the same/different source propositions.

Usage

```
two.level.lindley.LR(control, recovered, background)
```

Arguments

`control`	a `compitem` object with the control item information
`recovered`	a `compitem` object with the recovered item information
`background`	a `compcovar` object with the population information

Details

Does the likelihood ratio calculations for a two-level model assuming that the between item distribution is univariate normal. This function is taken from the approach devised by Denis Lindley in his 1977 paper (details below) and represents the progenitor of all the functions in this package.

Value

an estimate of the likelihood ratio

Author(s)

David Lucy

References

Lindley, D. (1977) A problem in forensic Science. *Biometrika*: **64**; 207-213.

Examples

```
# load Greg Zadora's glass data
data(glass)

# calculate a compcovar object based upon dat
# using K
Z = two.level.components(glass, 7, 1)

# calculate a compitem object representing the control item
control = two.level.comparison.items(glass[1:6,], 7)

# calculate a compitem object representing the recovered item
# known to be from the same item (item 1)
recovered.1 = two.level.comparison.items(glass[7:12,], 7)

# calculate a compitem object representing the recovered item
# known to be from a different item (item 2)
recovered.2 = two.level.comparison.items(glass[19:24,], 7)


# calculate the likelihood ratio for a known
# same source comparison - should be 6.323941
# This value is 6.323327 in this version and in the last version written by David (1.0-4)
lr.1 = two.level.lindley.LR(control, recovered.1, Z)
lr.1

# calculate the likelihood ratio for a known
# different source comparison - should be 0.004422907
# This value is 0.004421978 in this version and the last version written by David (1.0-4)
lr.2 = two.level.lindley.LR(control, recovered.2, Z)
lr.2
```

---

two.level.normal.LR *Likelihood ratio calculation - normal*

Description

Takes a `compitem` object which represents some control item, and a `compitem` object which represents a recovered item, then uses information from a `compcovar` object, which represents the information from the population, to calculate a likelihood ratio as a measure of the evidence given by the observations for the same/different source propositions.

Usage

```
two.level.normal.LR(control, recovered, background)
```

Arguments

`control`	a `compitem` object with the control item information
`recovered`	a `compitem` object with the recovered item information
`background`	a `compcovar` object with the population information

Details

Does the likelihood ratio calculations for a two-level model assuming that the between item distribution is uni/multivariate normal.

Value

an estimate of the likelihood ratio

Author(s)

Agnieszka Martyna and David Lucy

References

Aitken, C.G.G. & Lucy, D. (2004) Evaluation of trace evidence in the form of multivariate data. *Applied Statistics*: **53**(1); 109-122.

Examples

```
# load Greg Zadora's glass data
data(glass)

# calculate a compcovar object based upon glass
# using K, Ca and Fe - warning - could take time
# on slower machines
Z <- two.level.components(glass, c(7,8,9), 1)

# calculate a compitem object representing the control item
control <- two.level.comparison.items(glass[1:6,], c(7,8,9))

# calculate a compitem object representing the recovered item
# known to be from the same item (item 1)
recovered.1 <- two.level.comparison.items(glass[7:12,], c(7,8,9))
```

```
# calculate a compitem object representing the recovered item
# known to be from a different item (item 2)
recovered.2 <- two.level.comparison.items(glass[19:24,], c(7,8,9))


# calculate the likelihood ratio for a known
# same source comparison - should be 51.16539
# This value is 51.14243 in this version and the last version David wrote (1.0-4)
lr.1 <- two.level.normal.LR(control, recovered.1, Z)
lr.1
# calculate the likelihood ratio for a known
# different source comparison - should be 0.02901532
# This vsalue is 0.02899908 in this version and the last version David wrote (1.0-4)
lr.2 <- two.level.normal.LR(control, recovered.2, Z)
lr.2
```

Index