

Package ‘comparison’

August 25, 2023

Version 1.0.8

Date 2023-08-22

Title Multivariate Likelihood Ratio Calculation and Evaluation

Encoding UTF-8

Description Functions for calculating and evaluating likelihood ratios from uni/multivariate continuous observations.

Depends R (>= 3.5.0)

Imports isotone, CVglasso, methods

License GPL (>= 2)

URL <https://github.com/jmcurran/comparison>

LazyLoad yes

RoxygenNote 7.2.3

BugReports <https://github.com/jmcurran/comparison/issues>

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 2023-08-25 15:30:16 UTC

R topics documented:

calc.ece	2
calcLR	3
calibrate.set	4
glass	5
logistic.apply.calibration	7
logistic.calibrate.get.model	8
logistic.calibrate.set	9

makeCompItem	10
makeCompVar	11
plot.ece	12
print.compitem	13
two.level.comparison.items	13
two.level.components	14
two.level.density.LR	15
two.level.lindley.LR	16
two.level.normal.LR	17

Index	20
--------------	-----------

calc.ece	<i>Empirical cross-entropy (ECE) calculation</i>
----------	--

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

Ramos, D. & Gonzalez-Rodriguez, J. (2008) Cross-entropy analysis of the information in forensic speaker recognition; IEEE Odyssey. 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
```

calcLR

Calculate the likelihood ratio

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 (LR) as a measure of the evidence given by the observations for the same/different source propositions.

Usage

```
calcLR(control, recovered, background, method = c("mvn", "kde", "lindley"))
```

Arguments

<code>control</code>	a <code>compitem</code> object with the control item information.
<code>recovered</code>	a <code>compitem</code> object with the recovered item information.
<code>background</code>	a <code>compcovar</code> object with the population information.
<code>method</code>	a choice of the method used to calculate the LR. Presently there are three methods, "mvn" - multivariate normal approximation, "kde" - (multivariate) kernel density estimates and "lindely" which uses the method published by Lindley (1977).

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

```
data(glass)

controlMeasurements = subset(glass, item == "s1")
control = makeCompItem(item ~ logK0 + logCa0 + logFe0,
                      data = controlMeasurements[1:6,])
recovered.1 = makeCompItem(item ~ logK0 + logCa0 + logFe0,
                          data = controlMeasurements[7:12,])
recoveredMeasurements = subset(glass, item == "s2")
recovered.2 = makeCompItem(item ~ logK0 + logCa0 + logFe0,
                          data = recoveredMeasurements[7:12,])

background = makeCompVar(item ~ logK0 + logCa0 + logFe0, data = glass)

## Same source comparison using a multivariate normal (MVN) approximation
calcLR(control, recovered.1, background)

## Same source comparison using a multivariate kernel density estimate (MVK) approximation
calcLR(control, recovered.1, background, "kde")

## Different source comparison using a multivariate normal (MVN) approximation
calcLR(control, recovered.2, background)

## Different source comparison using a multivariate kernel density estimate (MVK) approximation
calcLR(control, recovered.2, background, "kde")
```

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 violation `loglr()` function for Matlab.

Usage

```
calibrate.set(
  LR.ss,
  LR.ds,
  method = c("raw", "laplace"),
  ties = c("none", "primary", "secondary", "tertiary")
)
```

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"
ties	method to solve ties in the predictors list, either "none" (not solved) or "primary", "secondary" or "tertiary" (passed to the <code>isotone::gpava()</code> function)

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

Ramos, D. & Gonzalez-Rodriguez, J. (2008) Cross-entropy analysis of the information in forensic speaker recognition; IEEE Odyssey.

de Leeuw, J. & Hornik, K. & Mair, P., (2009), Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods, <https://www.jstatsoft.org/article/view/v032i05>

See Also

[isotone::gpava\(\)](#), [calc.ece\(\)](#)

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.

`logistic.apply.calibration`*Calculate the calibrated LRs with the model precomputed*

Description

This function perform the logistic calibration on the provided data. In the context of likelihood ratios, the 'ideal' value for the LR is Infinity for the same source dataset, and 0 for the different-sources dataset. The 'post' values are fixed to 1 for the same source and 0 for the same different-sources datasets (corresponding to the posterior probability $P(H_{ss|E})$).

Usage

```
logistic.apply.calibration(LR, model)
```

Arguments

LR	a vector of likelihood ratios to be calibrated (raw values).
model	a <code>logistic.calibrate.set()</code> fitted model to be applied. This variable can be the return of the <code>logistic.calibrate.set()</code> or the <code>logistic.calibrate.set()\$fit</code> variable.

Value

a list with the calibrated LR values

Author(s)

Marco De Donno

See Also

[logistic.calibrate.set\(\)](#)

Examples

```
# the list of LRs for the same source proposition
LR.same = c(0.5, 2, 4, 6, 8, 10)
# the list of LRs for the different source proposition
LR.different = c(0.2, 0.4, 0.6, 0.8, 1.1)
# compute the logistic calibration on the data
model = logistic.calibrate.get.model(LR.same, LR.different)
# the list of news LRs (to be calibrated)
LR.unknown = c(0.6, 0.7, 1.2, 5)
# compute the calibrated LRs for the list with the model
logistic.apply.calibration(LR.unknown, model)
```

```
logistic.calibrate.get.model
```

Compute and returns the logistic regression for a dataset

Description

Compute and returns the logistic regression for a dataset

Usage

```
logistic.calibrate.get.model(LR.ss, LR.ds)
```

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

Value

a list with multiple items:

coefficients coefficients of the fitted model

prior.odds prior odds for the input data

Author(s)

Marco De Donno

See Also

[logistic.apply.calibration\(\)](#)

Examples

```
# the list of LRs for the same source proposition
LR.same = c(0.5, 2, 4, 6, 8, 10)
# the list of LRs for the different source proposition
LR.different = c(0.2, 0.4, 0.6, 0.8, 1.1)
# compute the logistic calibration on the data
logistic.calibrate.get.model(LR.same, LR.different)
```

`logistic.calibrate.set`*Calculate the calibrated set of LRs with the logistic regression*

Description

Calculate the calibrated set of LRs with the logistic regression

Usage

```
logistic.calibrate.set(LR.ss, LR.ds)
```

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

Value

a list with multiple items:

prior.odds prior odds for the input data

coefficients coefficients of the fitted model

data The input and calibrated data

LR.cal.ss The calibrated data for the same source list

LR.cal.ds The calibrated data for the different-sources list

Author(s)

Marco De Donno

See Also

[logistic.apply.calibration\(\)](#)

Examples

```
LR.same = c(0.5, 2, 4, 6, 8, 10)      # the list of LRs for the same source proposition
LR.different = c(0.2, 0.4, 0.6, 0.8, 1.1) # the list of LRs for the different source proposition
logistic.calibrate.set(LR.same, LR.different) # compute the logistic calibration on the data
```

makeCompItem *Create a compitem object.*

Description

This function creates a `compitem` from a set of observations on items to be deemed control, or a recovered, items. For example, a set of elemental concentration measurements on a sample of glass fragments taken from a crime scene source such as a window.

Usage

```
makeCompItem(x, ...)  
  
## S3 method for class 'formula'  
makeCompItem(x, data = NULL, ...)
```

Arguments

<code>x</code>	a <code>matrix</code> or <code>data.frame</code> or a <code>formula</code>
<code>...</code>	other arguments that may be passed to the function.
<code>data</code>	if <code>x</code> is a <code>formula</code> , then the user must supply a <code>data.frame</code> containing the observations.

Value

an object of class `compitem`

Methods (by class)

- `makeCompItem(formula)`: Create a `compitem` object using a `formula`.

Author(s)

David Lucy and James Curran

Examples

```
# load Greg Zadora's glass data  
data(glass)  
  
# calculate a compitem object representing the control item  
controlMeasurements = subset(glass, item == "s1", select = c(logK0, logCa0, logFe0))  
control = makeCompItem(controlMeasurements)  
  
# example using the formula interface  
controlMeasurements = subset(glass, item == "s1")  
control = makeCompItem(item ~ logK0 + logCa0 + logFe0, data = controlMeasurements)
```

`makeCompVar`*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

```
makeCompVar(x, ...)  
  
## Default S3 method:  
makeCompVar(x, item.column, ...)  
  
## S3 method for class 'formula'  
makeCompVar(x, data = NULL, ...)
```

Arguments

<code>x</code>	a matrix, or <code>data.frame</code> , of observations, with cases in rows, and properties as columns, or a formula.
<code>...</code>	other arguments.
<code>item.column</code>	an integer indicating which column gives the item.
<code>data</code>	if <code>x</code> is a formula, then the user must supply a <code>data.frame</code> containing the observations.

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`

Methods (by class)

- `makeCompVar(default)`: Create a `compvar` object using a formula.
- `makeCompVar(formula)`: Create a `compvar` object using a formula.

Author(s)

David Lucy and James Curran

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
background = subset(glass, select = c(item, logK0, logCa0, logFe0))
Z1 = makeCompVar(background, 1)

# Use the formula interface
Z2 = makeCompVar(item ~ logK0 + logCa0 + logFe0, data = glass)
```

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\(\)](#)

print.compitem	<i>S3 method for class compitem</i>
----------------	-------------------------------------

Description

S3 method for class compitem

Usage

```
## S3 method for class 'compitem'  
print(x, ...)
```

Arguments

x	an object of class compitem created by makeCompItem()
...	further arguments passed to or from other methods.

two.level.comparison.items	<i>Create a compitem object.</i>
----------------------------	----------------------------------

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

Note:

This function is deprecated and will eventually be replaced by [makeCompItem\(\)](#).

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 glass
# 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

<code>control</code>	a <code>compitem</code> object with the control item information.
<code>recovered</code>	a <code>compitem</code> object with the recovered item information.
<code>background</code>	a <code>compcovar</code> 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

<code>control</code>	a <code>compitem</code> object with the control item information
<code>recovered</code>	a <code>compitem</code> object with the recovered item information
<code>background</code>	a <code>compcovar</code> 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 value 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

* datasets

glass, 5

calc.ece, 2

calc.ece(), 5, 12

calcLR, 3

calibrate.set, 4

calibrate.set(), 3

glass, 5

isotone::gpava(), 3, 5

logistic.apply.calibration, 7

logistic.apply.calibration(), 8, 9

logistic.calibrate.get.model, 8

logistic.calibrate.set, 9

logistic.calibrate.set(), 7

makeCompItem, 10

makeCompItem(), 13

makeCompVar, 11

plot.ece, 12

print.compitem, 13

two.level.comparison.items, 13

two.level.components, 14

two.level.density.LR, 15

two.level.lindley.LR, 16

two.level.normal.LR, 17