Package 'molic'

June 2, 2021

Title Multivariate Outlier Detection in Contingency Tables

Version 2.0.3

Description Outlier detection in, possibly high-dimensional, categorical data following Mads Lindskou et al. (2019) <doi:10.1111/sjos.12407>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp

Imports Rcpp, doParallel, foreach, ggplot2, ggridges, ess

Suggests knitr, rmarkdown, pander, dplyr, testthat, igraph

VignetteBuilder knitr

RoxygenNote 7.1.1

SystemRequirements C++11

URL https://github.com/mlindsk/molic

BugReports https://github.com/mlindsk/molic/issues

NeedsCompilation yes

Author Mads Lindskou [aut, cre]

Maintainer Mads Lindskou <mads@math.aau.dk>

Repository CRAN

Date/Publication 2021-06-02 15:40:02 UTC

R topics documented:

`molic-package` *molic: Multivariate Ouliterdetection In Contingency Tables*

Description

Outlier detection in, possibly highdimensional, categorical data following Mads Lindskou et al. (2019) <doi:10.1111/sjos.12407>.

Author(s)

Maintainer: Mads Lindskou <mads@math.aau.dk>

See Also

Useful links:

- https://github.com/mlindsk/molic
- Report bugs at https://github.com/mlindsk/molic/issues

`cdf` *Empirical distribution function*

Description

The empirical cdf of `T(Y)`

Usage

```
cdf(x, ...)

## S3 method for class 'outlier_model'
cdf(x, ...)
```

Arguments

x	A outlier_model object
...	Not used (for S3 compatibility)

Value

The cumulative distribution of deviance test statistic of x

---

critval *Critical value*

Description

Calculate the critical value for test statistic under H_0

Usage

```
critval(m, alpha = 0.05)

## S3 method for class 'outlier_model'
critval(m, alpha = 0.05)
```

Arguments

m	A outlier_model object
alpha	Significance level (between 0 and 1)

Details

The value dz can be obtained used the deviance function.

Value

The critical value in the distribution of deviance test statistic of m

See Also

deviance

derma *Dermatology Database*

Description

We have removed 8 observations with missing values. Data contains 12 clinical attributes and 21 histopathological attributes. The age attribute has been discretized. The class variable has six levels; each describing a skin disease.

Usage

```
derma
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 358 rows and 35 columns.

References

https://archive.ics.uci.edu/ml/datasets/dermatology

deviance *Calculate deviance*

Description

This function calculates the affine value `T(y)` of `-2 log` likelihood-ratio statistic which is also called the deviance

Usage

```
deviance(x, y, ...)

## S3 method for class 'outlier_model'
deviance(x, y, ...)

## S3 method for class 'mixed_outlier'
deviance(x, y, ...)
```

Arguments

`x`	A `outlier_model` object
`y`	An observation (named character vector). If `x` is of class `mixed_outlier` it should be a `data.frame` with two rows.
`...`	Not used (for S3 compatibility)

Value

The deviance test statistic of y based on the model x

`fit_mixed_outlier` *Mixed Outlier Test*

Description

A function for outlier detection with mixed, but independen, information

Usage

```
fit_mixed_outlier(m1, m2)
```

Arguments

m1	An object returned from `fit_outlier`
m2	An object returned from `fit_outlier`

Details

It is assumed that the input data to `m1` and `m2` holds information about the same observation in corresponding rows. Thus, the two datasets must also be of same dimension.

Value

An object of type `mixed_outlier` with `novelty` or `outlier` as child classes. These are used for different purposes. See `fit_outlier`.

See Also

`fit_outlier`, `fit_multiple_models`, `outliers`, `pval`, `deviance`

Examples

```
library(dplyr)
library(ess)  # for fit_components
set.seed(7)   # for reproducibility

## Data

# The components - here microhaplotypes
haps <- tgp_haps[1:5] # only a subset of data is used to exemplify
dat <- tgp_dat %>%
       select(pop_meta, sample_name, all_of(unname(unlist(haps))))

# All the Europeans
```

```
eur <- dat %>%
  as_tibble() %>%
  filter(pop_meta == "EUR")

# Extracting the two databases for each copy of the chromosomes
eur_a <- eur %>%
  filter(grepl("a$", sample_name)) %>%
  select(-c(1:2))

eur_b <- eur %>%
  filter(grepl("b$", sample_name)) %>%
  select(-c(1:2))


# Fitting the interaction graphs on the EUR data
ga <- fit_components(eur_a, comp = haps, trace = FALSE)
gb <- fit_components(eur_b, comp = haps, trace = FALSE)

## ------------------------------------------------------------
##                        EXAMPLE 1
##   Testing which observations within data are outliers
## ------------------------------------------------------------

# Only 500 simulations is used here to exeplify
# The default number of simulations is 10,000
m1 <- fit_outlier(eur_a, ga, nsim = 500) # consider using more cores (ncores argument)
m2 <- fit_outlier(eur_b, gb, nsim = 500) # consider using more cores (ncores argument)
m  <- fit_mixed_outlier(m1, m2)
print(m)
plot(m)

outs <- outliers(m)
eur_a_outs <- eur_a[which(outs), ]
eur_b_outs <- eur_b[which(outs), ]

# Retrieving the test statistic for individual observations
x1 <- rbind(eur_a_outs[1, ], eur_b_outs[1, ])
x2 <- rbind(eur_a[1, ], eur_b[1, ])
dev1 <- deviance(m, x1) # falls within the critical region in the plot (the red area)
dev2 <- deviance(m, x2) # falls within the acceptable region in the plot

dev1
dev2

# Retrieving the pvalues
pval(m, dev1)
pval(m, dev2)




## ------------------------------------------------------------
##                        EXAMPLE 2
```

```
##      Testing if a new observation is an outlier
## ------------------------------------------------------------

# Testing if an American is an outlier in Europe
amr <- dat %>%
  as_tibble() %>%
  filter(pop_meta == "AMR")

z1  <- amr %>%
  filter(grepl("a$", sample_name)) %>%
  select(unname(unlist(haps))) %>%
  slice(1) %>%
  unlist()

z2  <- amr %>%
  filter(grepl("b$", sample_name)) %>%
  select(unname(unlist(haps))) %>%
  slice(1) %>%
  unlist()

# Only 500 simulations is used here to exemplify
# The default number of simulations is 10,000
m3 <- fit_outlier(eur_a, ga, z1, nsim = 500) # consider using more cores (ncores argument)
m4 <- fit_outlier(eur_b, gb, z2, nsim = 500) # consider using more cores (ncores argument)
m5 <- fit_mixed_outlier(m3, m4)
print(m5)
plot(m5)
```

`fit_multiple_models` *Fit Multiple Models*

Description

Conduct multiple novelty tests for a new observation

Usage

```
fit_multiple_models(
  A,
  z,
  response,
  alpha = 0.05,
  type = "fwd",
  q = 0.5,
  comp = NULL,
  nsim = 10000,
```

```
  ncores = 1,
  trace = TRUE,
  validate = TRUE
)
```

Arguments

A	A character matrix or data.frame
z	Named vector. Same names as colnames(A) but without the class variable
response	A character with the name of the class variable of interest
alpha	The significance level
type	Character ("fwd", "bwd", "tree" or "tfwd") - the type of interaction graph to be used
q	Penalty term in the stopping criterion when fitting the interaction graph (0 = AIC and 1 = BIC)
comp	A list with character vectors. Each element in the list is a component in the graph (using expert knowledge)
nsim	Number of simulations
ncores	Number of cores to use in parallelization
trace	Logical indicating whether or not to trace the procedure
validate	Logical. If true, it checks if A has only single character values and converts it if not.

Value

An object of type multiple_models; a list of of novely objects from which one can query pvalues etc. for outlierdetection.

See Also

fit_outlier, fit_mixed_outlier

Examples

```
library(dplyr)
set.seed(1)

# A patient with psoriasis
z <- unlist(derma[2, 1:10])

d <- derma[, c(names(z), "ES")] %>%
     filter(ES %in% c("chronic dermatitis", "psoriasis"))

m <- fit_multiple_models(d, z, "ES", nsim = 1000, trace = FALSE, validate = FALSE)

plot(m)
print(m)
```

`fit_outlier` *Outlier detection*

Description

Detecting outliers within a dataset or test if a new (novel) observation is an outlier.

Usage

```
fit_outlier(
  A,
  adj,
  z = NULL,
  alpha = 0.05,
  nsim = 10000,
  ncores = 1,
  validate = TRUE
)
```

Arguments

`A`	Character matrix or data.frame. All values must be limited to a single character.
`adj`	Adjacency list or `gengraph` object of a decomposable graph. See package `ess` for `gengraph` objects.
`z`	Named vector (same names as `colnames(A)`) or `NULL`. See details. Values must be limited to a single character.
`alpha`	Significance level
`nsim`	Number of simulations
`ncores`	Number of cores to use in parallelization
`validate`	Logical. If true, it checks if `A` only has single character values and converts it if not.

Details

If the goal is to detect outliers within `A` set `z` to `NULL`; this procedure is most often just referred to as outlier detection. Once `fit_outlier` has been called in this situation, one can exploit the `outliers` function to get the indicies for which observations in `A` that are outliers. See the examples.

On the other hand, if the goal is test if the new unseen observation `z` is an outlier in`A`, then supply a named vector to `z`.

All values must be limited to a single character representation; if not, the function will internally convert to one such representation. The reason for this, is a speedup in runtime performance. One can also use the exported function `to_chars` on `A` in advance and set `validate` to `FALSE`.

The `adj` object is most typically found using `fit_graph` from the `ess` package. But the user can supply an adjacency list, just a named `list`, of their own choice if needed.

Value

A `outlier_model` object with either `novelty` or `outlier` as child classes. These are used for different purposes. See the details

See Also

`fit_mixed_outlier`, `fit_multiple_models`, `outliers`, `pval`, `deviance`

Examples

```
library(dplyr)
library(ess)  # For the fit_graph function
set.seed(7)   # For reproducibility

# Psoriasis patients
d <- derma %>%
  filter(ES == "psoriasis") %>%
  select(1:20) %>% # only a subset of data is used to exemplify
  as_tibble()

# Fitting the interaction graph
# see package ess for details
g <- fit_graph(d, trace = FALSE)
plot(g)

# ------------------------------------------------------------
#                         EXAMPLE 1
#    Testing which observations within d are outliers
# ------------------------------------------------------------

# Only 500 simulations is used here to exeplify
# The default number of simulations is 10,000
m1 <- fit_outlier(d, g, nsim = 500)
print(m1)
outs  <- outliers(m1)
douts <- d[which(outs), ]
douts

# Notice that m1 is of class 'outlier'. This means, that the procedure has tested which
# observations _within_ the data are outliers. This method is most often just referred to
# as outlier detection. The following plot is the distribution of the test statistic. Think
# of a simple t-test, where the distribution of the test statistic is a t-distribution.
# In order to conclude on the hypothesis, one finds the critical value and verify if the
# test statistic is greater or less than this.

# Retrieving the test statistic for individual observations
x1 <- douts[1, ] %>% unlist()
x2 <- d[1, ] %>% unlist()
dev1 <- deviance(m1, x1) # falls within the critical region in the plot (the red area)
dev2 <- deviance(m1, x2) # falls within the acceptable region in the plot
```

```
dev1
dev2

# Retrieving the pvalues
pval(m1, dev1)
pval(m1, dev2)

# ------------------------------------------------------------
#                        EXAMPLE 2
#         Testing if a new observation is an outlier
# ------------------------------------------------------------

# An observation from class "chronic dermatitis"
z <- derma %>%
  filter(ES == "chronic dermatitis") %>%
  select(1:20) %>%
  slice(1) %>%
  unlist()

# Test if z is an outlier in class "psoriasis"
# Only 500 simulations is used here to exeplify
# The default number of simulations is 10,000
m2 <- fit_outlier(d, g, z, nsim = 500)
print(m2)
plot(m2) # Try using more simulations and the complete derma data

# Notice that m2 is of class 'novelty'. The term novelty detection
# is sometimes used in the litterature when the goal is to verify
# if a new unseen observation is an outlier in a homogen dataset.

# Retrieving the test statistic and pvalue for z
dz <- deviance(m2, z)
pval(m2, dz)
```

`mean.outlier_model` *Mean*

Description

Estimated mean of deviance statistic `T(Y)`

Usage

```
## S3 method for class 'outlier_model'
mean(x, ...)
```

Arguments

`x`	A `outlier_model` object
`...`	Not used (for S3 compatibility)

Value

The mean of the deviance test statistic of x

---

outliers *Detect Outliers*

Description

Find the outliers some given data based on an outlier model

Usage

```
outliers(x, alpha = 0.05)

## S3 method for class 'outlier'
outliers(x, alpha = 0.05)

## S3 method for class 'mixed_outlier'
outliers(x, alpha = 0.05)
```

Arguments

x	A outlier object
alpha	Sigficance level

Value

Vector of logicals referring to the indicies in the data used to call x for which the observations are outliers.

---

plot.multiple_models *Plot Deviance of Multiple Models*

Description

A plot method to show the approximated deviance distribution of multiple models

Usage

```
## S3 method for class 'multiple_models'
plot(x, sig_col = "#FF0000A0", ...)
```

Arguments

x	A multiple_models object returned from a called to fit_multiple_models
sig_col	Color of the significance level area (default is red)
...	Extra arguments. See details.

Details

The dotted line represents the observed deviance of the observation under the hypothesis and the colored (red is default) area under the graph represents the significance level. Thus, if the dotted line is to the left of the colored area, the hypothesis that the observation is an outlier cannot be rejected. Notice however, if there is no dotted line, this simply means, that the observed deviance is larger than all values and it would disturb the plot if included.

No extra arguments ... are implement at the moment.

Value

No return value, called for side effects

---

plot.outlier_model *Plot Distribution of Test Statistic*

Description

A plot method to show the approximated distribution of the deviance test statistic

Usage

```
## S3 method for class 'outlier_model'
plot(x, sig_col = "#FF0000A0", ...)
```

Arguments

x	An object returned from fit_outlier
sig_col	Color of the significance level area (default is red)
...	Not used. For S3 compatability.

Details

The dotted line represents the observed test statistic of z and the colored (red is default) area under the graph represents the significance level.

Thus, if z is supplied and the dotted line is to the left of the colored area, the hypothesis that the observation is an outlier cannot be rejected. Notice however, if there is no dotted line, this simply means, that the observed test statistic is larger than all values and it would disturb the plot if included.

Value

No return value, called for side effects

`print.outlier_model` *Print outlier model*

Description

A print method for `outlier_model` objects

Usage

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

Arguments

`x`	A `outlier_model` object
`...`	Not used (for S3 compatability)

Value

No return value, called for side effects

`pval` *P-value*

Description

Calculate the p-value for obtaining `ty_new` under `H_0`

Usage

```
pval(x, dz, ...)

## S3 method for class 'outlier_model'
pval(x, dz, ...)
```

Arguments

`x`	A `outlier_model` object
`dz`	The deviance of the observation `z`.
`...`	Not used (for S3 compatibility)

Details

The value dz can be obtained used the deviance function.

Value

The p-value of deviance test statistic of x

See Also

deviance

---

tgp_dat *A data frame with genetic data from the 1000 genomes project*

Description

The data consists of 2504 DNA profiles, each genotyped on 304 SNPs (binary variables). The data frame has 5008 rows, since each profile has two copies.

Usage

```
tgp_dat
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 5008 rows and 304 columns.

References

1000 Genomes Project

---

tgp_haps *A named list of character vectors.*

Description

Every element in the list is a character vector that forms a haplotype from the 1000 genomes project. If the list is unlisted, it should correspond to colnames of tgp_dat. In other words, tgp_haps is a "haplotype-grouping" of the variables in tgp_dat.

Usage

```
tgp_haps
```

Format

An object of class list of length 109.

References

1000 Genomes Project

`to_chars` *Convert discrete values into a single character representation*

Description

Convert all values in a data frame or matrix of characters to a single character representation

Usage

```
to_chars(x)
```

Arguments

`x`	Data frame or matrix of characters

Examples

```
d <- data.frame(x = c("11", "2"), y = c("2", "11"))
to_chars(d)
```

`variance` *Variance*

Description

Estimated variance of the deviance statistic `T(Y)`

Usage

```
variance(x)

## S3 method for class 'outlier_model'
variance(x, ...)
```

Arguments

`x`	A `outlier_model` object
`...`	Not used (for S3 compatibility)

Value

The variance of the deviance test statistic of `x`

Index