Package 'MultiFit'

March 17, 2019

Type Package

Title Multivariate Multiscale Framework for Independence Tests

Version 1.0.0

Author S. Gorsky, L. Ma

Maintainer S. Gorsky <s.gorsky@duke.edu>

Description Test for independence of two random vectors, learn and report the dependency structure. For more information, see Gorsky and Ma (2018) <arXiv:1806.06777>.

License CC0

Imports Rcpp (>= 0.12.17), data.table

LinkingTo Rcpp, RcppArmadillo

Suggests png, qgraph, knitr, rmarkdown

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-03-17 18:43:34 UTC

R topics documented:

multiFit *Multiscale Multivariate Framework for Independence Tests*

Description

Perform multiscale test of independence for multivariate vectors. See vignettes for further examples.

Usage

```
multiFit(xy, x = NULL, y = NULL, p_star = NULL, R_max = NULL,
  R_star = 1, rank.transform = TRUE, test.method = "Fisher",
  correct = TRUE, min.tbl.tot = 25L, min.row.tot = 10L,
  min.col.tot = 10L, p.adjust.methods = c("H", "Hcorrected", "MH"),
  compute.all.holm = TRUE, cutoff = 0.05, top.max.ps = 4L,
  return.all.pvs = TRUE, save.all.pvs = FALSE, all.pvs.fname = NULL,
  uv.approx.null = FALSE, uv.exact.null = FALSE,
  uv.null.sim = 10000L, plot.marginals = FALSE, rk = FALSE, M = 10,
  verbose = FALSE)
```

Arguments

xy	A list, whose first element corresponds to the matrix x as below, and its second element corresponds to the matrix y as below. If xy is not specified, x and y need to be assigned.
x	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
y	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
p_star	Numeric, cuboids associated with tests whose p-value is below p_star will be halved and further tested.
R_max	A positive integer (or Inf), the maximal number of resolutions to scan (algorithm will stop at a lower resolution if all tables in it do not meet the criteria specified at min.tbl.tot, min.row.tot and min.col.tot)
R_star	A positive integer, if set to an integer between 0 and R_max, all tests up to and including resolution R_star will be performed (algorithm will stop at a lower resolution than requested if all tables in it do not meet the criteria specified at min.tbl.tot, min.row.tot and min.col.tot). For higher resolutions only the children of tests with p-value lower than p_star will be considered.
rank.transform	Logical, if TRUE, marginal rank transform is performed on all margins of x and y. If FALSE, all margins are scaled to 0-1 scale. When FALSE, the average and top statistics of the negative logarithm of the p-values are only computed for the univariate case.
test.method	String, choose "Fisher" for Fisher's exact test (slowest), "chi.sq" for Chi-squared test, "LR" for likelihood-ratio test and "norm.approx" for approximating the hypergeometric distribution with a normal distribution (fastest).

correct	Logical, if TRUE compute mid-p corrected p-values for Fisher's exact test, or Yates corrected p-values for the Chi-squared test, or Williams corrected p-values for the likelihood-ratio test.
min.tbl.tot	Non-negative integer, the minimal number of observations per table below which a p-value for a given table will not be computed.
min.row.tot	Non-negative integer, the minimal number of observations for row totals in the 2x2 contingency tables below which a contingency table will not be tested.
min.col.tot	Non-negative integer, the minimal number of observations for column totals in the 2x2 contingency tables below which a contingency table will not be tested.
p.adjust.methods	String, choose between "H" for Holm, "Hcorrected" for Holm with the correction as specified in correct, or "MH" for Modified-Holm (for Fisher's exact test only). See documentation for further details.
compute.all.holm	Logical, if FALSE, only global p-value is computed (may be faster, especially when Modified-Holm correction is used). If TRUE adjusted p-values are computed for all tests.
cutoff	Numerical between 0 and 1, an upper limit for the p-values that are to be adjusted (the lower the cutoff - the fewer computations are required for the Modified Holm method).
top.max.ps	Positive integer, report the mean of the top top.max.ps order statistics of the negative logarithm of all p-values.
return.all.pvs	Logical, if TRUE, a data frame with all p-values is returned.
save.all.pvs	Logical, if TRUE a data frame with all p-values is saved to a RData file named according to all.pvs.fname
all.pvs.fname	String, file name to which all p-values are saved if save.all.pvs is TRUE.
uv.approx.null	Logical, in a univariate case, if TRUE and the testing method is either Fisher's exact test or the normal approximation of the hypergeometric distribution, an approximate null distribution for the global test statistics is simulated. See documentation for further details.
uv.exact.null	Logical, in a univariate case, if TRUE and the testing method is either Fisher's exact test or the normal approximation of the hypergeometric distribution, an exact null distribution for the global test statistics is simulated. See documentation for further details.
uv.null.sim	Positive integer, the number of simulated values to be computed in a univariate case when an exact or approximate null distribution is simulated.
plot.marginals	Logical, if TRUE plots the marginal scatter plots between all pairs of margins of x and y, before and after rank transforming or scaling them.
rk	Logical, if FALSE, select only tests with p-values more extreme than p_star to halve and further test. FWER control guaranteed. If TRUE, choose at each resolution the M tests with the most extreme p-values to further halve and test.
M	A positive integer (or Inf), the number of top ranking tests to continue to split at each resolution. FWER control not guaranteed for this method.
verbose	Logical.

Value

test.stats, a named numerical vector containing the test statistics for the global null hypothesis (i.e. x independent of y)

p.values, a named numerical vector containing the p-values of for the global null hypothesis (i.e. x independent of y). These are not computed if p.adjust.methods is NULL.

pvs, a data frame that contains all p-values and adjusted p-values that are computed. Returned if return.all.pvs is TRUE.

all, a nested list. Each entry is named and contains data about a resolution that was tested. Each resolution is a list in itself, with cuboids, a summary of all tested cuboids in a resolution, tables, a summary of all 2x2 contingency tables in a resolution, pv, a numerical vector containing the p-values from the tests of independence on 2x2 contingency table in tables that meet the criteria defined by min.tbl.tot, min.row.tot and min.col.tot. The length of pv is equal to the number of rows of tables. pv.correct, similar to the above pv, corrected p-values are computed and returned when correct is TRUE. rank.tests, logical vector that indicates whether or not a test was ranked among the top M tests in a resolution. The length of rank.tests is equal to the number of rows of tables. parent.cuboids, an integer vector, indicating which cuboids in a resolution are associated with the ranked tests, and will be further halved in the next higher resolution. parent.tests, a logical vector of the same length as the number of rows of tables, indicating whether or not a test was chosen as a parent test (same tests may have multiple children).

approx.nulls, in a univariate case, a list of numerical vectors whose values are the simulated approximate null values.

exact.nulls, in a univariate case, a list of numerical vectors whose values are the simulated theoretical null values.

Examples

```
set.seed(1)
n = 300
Dx = Dy = 2
x = matrix(0, nrow=n, ncol=Dx)
y = matrix(0, nrow=n, ncol=Dy)
x[,1] = rnorm(n)
x[,2] = runif(n)
y[,1] = rnorm(n)
y[,2] = sin(5*pi*x[,2]) + 1/5*rnorm(n)
fit = multiFit(x=x, y=y, verbose=TRUE)
w = multiSummary(x=x, y=y, fit=fit, alpha=0.0001)
```

---

multiSummary *Summary of significant tests*

Description

Provide a post-hoc summary of significant tests. See vignettes for further examples.

Usage

```
multiSummary(xy, x = NULL, y = NULL, fit, alpha = 0.05,
  only.rk = NULL, use.pval = NULL, plot.tests = TRUE, pch = NULL,
  rd = 2, plot.margin = FALSE)
```

Arguments

xy	A list, whose first element corresponds to the matrix x as below, and its second element corresponds to the matrix y as below. if xy is not specified, x and y need to be assigned.
x	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
y	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
fit	An object generated by multiFit.
alpha	Numeric, only tests with adjusted p-values less than alpha are presented in the output.
only.rk	Positive integer vector. Show only tests that are ranked according to only.rk and have adjusted p-value below alpha. If left as NULL, all tests with adjusted p-values less than alpha are presented in the output.
use.pval	String, choose between "H" (for Holm), "Hcorrected" (for Holm on corrected p-values) or "MH" for modified Holm. If left NULL, the order of preference is "MH", "Hcorrected" and then "H", according to which is present in the object fit.
plot.tests	Logical, plot the marginal scatter plots that are associated with the presented significant tests.
pch	Point style for plots. If left as NULL, a default combination of crosses and bullets is applied.
rd	Numeric, number of figures to round to when presenting ranges of variables.
plot.margin	Logical, plot the marginal scatter plot of the margins that are associated with each significant test, without highlighting which points are conditioned on and are in the discretized 2x2 contingency table.

Value

List whose elements are significant.tests, a data frame that summarizes the main features of the tests and their overall ranking by p-value and original.scale.cuboids, a list whose number of elements is equal to the number of significant tests (the same number of rows of the data frame significant.tests). Each element corresponds to a test and is a list whose elements are the marginal ranges of the associated cuboid.

Examples

```
set.seed(1)
n = 300
Dx = Dy = 2
```

```
x = matrix(0, nrow=n, ncol=Dx)
y = matrix(0, nrow=n, ncol=Dy)
x[,1] = rnorm(n)
x[,2] = runif(n)
y[,1] = rnorm(n)
y[,2] = sin(5*pi*x[,2]) + 1/5*rnorm(n)
fit = multiFit(x=x, y=y, verbose=TRUE)
w = multiSummary(x=x, y=y, fit=fit, alpha=0.0001)
```

`multiTree` *Plot tree structure of tests on 2x2 contingency tables*

Description

Plot a post-hoc tree of all tests or all significant tests on 2x2 discretized contingency tables. See vignettes for examples.

Usage

```
multiTree(xy, x = NULL, y = NULL, fit, show.all = FALSE,
  max.node.size = 5, min.node.size = 2.5, use.pval = NULL,
  images.path = NULL, node.name = "node", filename = NULL,
  filetype = "pdf")
```

Arguments

`xy`	A list (optional), whose first element corresponds to the matrix `x` as below, and its second element corresponds to the matrix `y` as below. if `xy` is not specified, `x` and `y` need to be assigned. If `xy`, `x` and `y` are missing or `NULL`, the tree nodes are blank. If `xy` or `x` and `y` are provided, nodes are `png` images of the marginal scatter plots that are associated with each test.
`x`	A matrix (optional), number of columns = dimension of random vector, number of rows = number of observations. If `xy`, `x` and `y` are missing or `NULL`, the tree nodes are blank. If `xy` or `x` and `y` are provided, nodes are `png` images of the marginal scatter plots that are associated with each test.
`y`	A matrix (optional), number of columns = dimension of random vector, number of rows = number of observations. If `xy`, `x` and `y` are missing or `NULL`, the tree nodes are blank. If `xy` or `x` and `y` are provided, nodes are `png` images of the marginal scatter plots that are associated with each test.
`fit`	An object generated by `multiFit`.
`show.all`	Logical. If `TRUE`, all tests are shown. If `FALSE` only tests who were ranked in each resolution amongst the top `M` ranking tests are shown. See `?multiFit` for an explanation about the parameter `M` and see documentation for further information.
`max.node.size`	Numeric. Maximal node size. All nodes are scaled between `min.node.size` and `max.node.size`, where larger nodes are associated smaller p-values of the corresponding tests on 2x2 contingency tables.

min.node.size	Numeric. Minimal node size. All nodes are scaled between min.node.size and max.node.size, where larger nodes are associated smaller p-values of the corresponding tests on 2x2 contingency tables.
use.pval	String, choose between "H" (for Holm), "Hcorrected" (for Holm on corrected p-values) or "MH" for modified Holm. If left NULL, the order of preference is "MH", "Hcorrected" and then "H", according to which is present in the object fit.
images.path	String, path to save png images of nodes to. If not specified, images are saved to tempdir().
node.name	String, prefix for file names for nodes pngs.
filename	String, file name for tree output. If left NULL, file name is prefixed by multiTree and ends with system time. See documentation of qgraph::qgraph for further information.
filetype	String, default is pdf, See documentation of qgraph::qgraph for further information.

Value

The main output of multiTree is a pdf file with the directed acyclic graph showing tests as nodes.

In addition, the function returns a list. Its elements are: qgraph.object, the graphical object generated by the qgraph function. See the qgraph package documentation for further details. qgraph.call, the call for the tree generating function. Arguments for the call: adj, the adjacency matrix, nodes.size, a numeric vector with the scaled sizes of the nodes, images, the file names of the nodes images (may be NULL), filename as passed to multiTree and passed over to qgraph, and filetype as passed to multiTree and passed over to qgraph.

Other elements of the returned list are pvs.attributes, the attributes summarizing the data and the tests performed as stored in fit, and n.nodes, the number of nodes.

---

permNullTest *Permutation null distribution*

Description

Simulate a permutation null distribution (per input data and testing parameters) for the global test statistics. See vignettes for examples of usage.

Usage

```
permNullTest(perm.null.sim = 10000L, xy, x = x, y = y, fit,
  parts = 4L, save.perm.null = FALSE, perm.null.fname = NULL,
  verbose = FALSE)
```

Arguments

perm.null.sim	Positive integer, the number of simulated values to be computed when a permutation null distribution is simulated.
xy	A list, whose first element corresponds to the matrix x as below, and its second element corresponds to the matrix y as below. If xy is not specified, x and y need to be assigned.
x	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
y	A matrix, number of columns = dimension of random vector, number of rows = number of observations.
fit	An object generated by multiFit.
parts	Positive integer, divide computation to four parts. Useful for getting a sense of progress (when verbose=TRUE) or for distributing memory requirement.
save.perm.null	Logical, if TRUE, save the permutation null into an RData file named perm.null.fname
perm.null.fname	String, file name to which to save an RData file containing the permutation null if save.perm.null=TRUE.
verbose	Logical.

---

uvApproxNull *Approximate univariate null distribution*

Description

In a univariate case, simulate an approximate null distribution for the global test statistics.

Usage

```
uvApproxNull(uv.null.sim, num.wins, top.max.ps = 4L, verbose = FALSE)
```

Arguments

uv.null.sim	Positive integer, the number of simulated values to be computed in a univariate case when an exact or approximate null distribution is simulated.
num.wins	Positive integer, the number of windows that are tested in each simulation.
top.max.ps	Positive integer, report the mean of the top top.max.ps order statistics of the negative logarithm of all p-values.
verbose	Logical.

Value

List of two numerical vectors for each of the global test statistics. Each such vector is of length uv.null.sim.

uvExactNull *Exact univariate null distribution*

Description

In a univariate case, simulate an exact null distribution for the global test statistics.

Usage

```
uvExactNull(uv.null.sim, test.method, correct = TRUE, col0.tot, row0.tot,
  grand.tot, top.max.ps = 4L, verbose = FALSE)
```

Arguments

uv.null.sim	Positive integer, the number of simulated values to be computed in a univariate case when an exact or approximate null distribution is simulated.
test.method	String, "Fisher" and "norm.approx" are applicable here.
correct	Logical, relates to Fisher's exact test, if TRUE the exact null is simulated for the mid-p corrected p-values.
col0.tot	Numerical vector, containing the column-0 totals of all 2x2 contingency tables for which simulated values are asked for. Has to be the same length as row0.tot and grand.tot.
row0.tot	Numerical vector, containing the row-0 totals of all 2x2 contingency tables for which simulated values are asked for. Has to be the same length as col0.tot and grand.tot.
grand.tot	Numerical vector, containing the totals of all 2x2 contingency tables for which simulated values are asked for. Has to be the same length as col0.tot and row0.tot.
top.max.ps	Positive integer, report the mean of the top top.max.ps order statistics of the negative logarithm of all p-values.
verbose	Logical.

Value

List of two numerical vectors for each of the global test statistics. Each such vector is of length uv.null.sim.

Index