Package 'BET'

October 12, 2022

Type Package

Title Binary Expansion Testing

Version 0.5.0

Depends R (>= 3.5.0)

Description Nonparametric detection of nonuniformity and dependence with Binary Expansion Testing (BET). See Kai Zhang (2019) BET on Independence, Journal of the American Statistical Association, 114:528, 1620-1637, <DOI:10.1080/01621459.2018.1537921>, Kai Zhang, Zhigen Zhai, and Wen Zhou. (2021). BEAUTY Powered BEAST, <arXiv:2103.00674> and Zhigen Zhao, Michael Baiocchi, Kai Zhang. SorBET: A Fast and Powerful Algorithm to Test Dependence of Variables.

License GPL

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.3)

LinkingTo Rcpp

RoxygenNote 7.1.1

NeedsCompilation yes

Author Wan Zhang [aut, cre],
Zhigen Zhao [aut],
Michael Baiocchi [aut],
Kai Zhang [aut]

Maintainer Wan Zhang <wanz63@live.unc.edu>

Repository CRAN

Date/Publication 2022-05-11 16:40:02 UTC

R topics documented:

BEAST *Binary Expansion Adaptive Symmetry Test*

Description

BEAST (Binary Expansion Adaptive Symmetry Test) is used for nonparametric detection of nonuniformity or dependence.

Usage

```
BEAST(
  X,
  dep,
  subsample.percent = 1/2,
  B = 100,
  unif.margin = FALSE,
  lambda = NULL,
  index = list(c(1:ncol(X))),
  method = "p",
  num = NULL
)
```

Arguments

X	a matrix to be tested.
dep	depth of the binary expansion for the BEAST.
subsample.percent	sample size for subsampling.
B	times of subsampling.
unif.margin	logicals. If TRUE the marginal distribution is known to be Uniform[0,1]. Default is FALSE, and empirical cdf transformation will be applied to each marginal distribution.
lambda	tuning parameter for soft-thresholding, default to be $sqrt(log(2^(p * dep))/(8 * n))$.
index	a list of indices. If provided, test the independence among two or more groups of variables. For example, index = list(c(1,2), c(3))) refers to test the independence between (X_1, X_2) and X_3. Default to be list(c(1:p)) to test if the data follow the multivariate uniform distribution over [0,1]^p, where p = ncol(X).

method	If "p", then compute null distribution with permutations. If "s", then compute null distribution with simulations. If "stat", only return interaction and BEAST Statistic.
num	number of permutations if method == "p" (default to be 100), or simulations if method == "s" (default to be 1000).

Value

Interaction	the most frequent interaction among all subsamples.
BEAST.Statistic	BEAST statistic.
Null.Distribution	simulated null distribution.
p.value	simulated p-value.

Examples

```
## Elapsed times 7.73 secs
## Measured in R 4.0.2, 32 bit, on a processor 3.3 GHz 6-Core Intel Core i5 under MacOS, 2021/9/8
## Not run:
x1 = runif(128)
x2 = runif(128)
y = sin(4*pi*(x1 + x2)) + 0.8*rnorm(128)
##test independence between (x1, x2) and y
BEAST(cbind(x1, x2, y), 3, index = list(c(1,2), c(3)))
##test mutual independence among x1, x2 and y
BEAST(cbind(x1, x2, y), 3, index = list(1, 2, 3))

##test bivariate uniformity
x1 = rbeta(128, 2, 4)
x2 = rbeta(128, 2, 4)
BEAST(cbind(x1, x2), 3)
##test multivariate uniformity
x1 = rbeta(128, 2, 4)
x2 = rbeta(128, 2, 4)
x3 = rbeta(128, 2, 4)
BEAST(cbind(x1, x2, x3), 3)

##with a known simulation
BEAST.null <- BEAST.null.simu(128, 3, 3,index = list(c(1,2), c(3)))
x1 = runif(128)
x2 = runif(128)
y = sin(4*pi*(x1 + x2)) + 0.8*rnorm(128)
BEAST.stat = BEAST(cbind(x1, x2, y), 3, index = list(c(1,2), c(3)),
    method = "stat")$BEAST.Statistic
mean(BEAST.stat<BEAST.null)  # p-value

## End(Not run)
```

BEAST.null.simu *BEAST Null Distribution*

Description

BEAST.null.simu gives a vector of the null distribution of the BEAST statistic.

Usage

```
BEAST.null.simu(
  n,
  p,
  dep,
  subsample.percent = 1/2,
  B = 100,
  lambda = NULL,
  index = list(c(1:p)),
  method = "p",
  num = NULL
)
```

Arguments

n	sample size.
p	dimension.
dep	depth of the binary expansion.
subsample.percent	sample size for subsampling.
B	times of subsampling.
lambda	tuning parameter for soft-thresholding, default to be $sqrt(log(2^(p * dep))/(8 * n))$.
index	a list of indices. If provided, test the independence among two or more groups of variables. For example, index = list(c(1,2), c(3))) refers to test the independence between (X_1, X_2) and X_3. Default to be list(c(1:p)) to test if the data follow the uniform distribution over [0,1]^p, where p = ncol(X).
method	If "p", then compute null distribution with permutations. If "s", then compute null distribution with simulations.
num	number of permutations if method == "p" (default to be 100), or simulations if method == "s" (default to be 1000).

Value

BEAST.null.simu returns a vector of length num.permutations that simulates the null distribution of the BEAST for given sample size n, dimension p, and depth D

Examples

```
## Elapsed times 2.4 secs
## Measured in R 4.0.2, 32 bit, on a processor 3.3 GHz 6-Core Intel Core i5 under MacOS, 2021/5/30.
## Not run: BEAST.null.simu(128, 2, 3)

## power study example
## Elapsed times 36.8 secs
## Measured in R 4.0.2, 32 bit, on a processor 3.3 GHz 6-Core Intel Core i5 under MacOS, 2021/5/30.
## Not run:
nperm = 100
nsim = 1000
BEAST.res = rep(0, nsim)
## simulate null distribution: only need once
BEAST.null.dist = BEAST.null.simu(128, 3, 3, index = list(c(1,2), c(3)), num = nperm)
for(i in 1:nsim){
  x1 = runif(128)
  x2 = runif(128)
  y = sin(4*pi*(x1 + x2)) + 0.8*rnorm(128)
  BEAST.stat = BEAST(cbind(x1, x2, y), 3, index = list(c(1,2), c(3)),
    method = "stat")$BEAST.Statistic
  BEAST.pvalue = sum(BEAST.null.dist >= BEAST.stat) / nperm
  BEAST.res[i] = BEAST.pvalue
}
## compute power
level = 0.1
power = mean(BEAST.res < level)


## End(Not run)
```

`bet.plot` *Plotting Binary Expansion Testing (2-dimensions)*

Description

`bet.plot` shows the cross interaction of the strongest asymmetry, which the BET returns with the rejection of independence null. This function only works for the test on two variables, that is, `X` can only have two columns. There are $2^{2dep} - 1$ nontrivial binary variables in the σ-field and $(2^d ep - 1)^2$ of them are cross interactions, whose positive regions are in plotted in white and whose negative regions are plotted in blue. `plot.bet` shows the cross interaction where the difference of number of observations in the positive and negative region is largest.

Usage

```
bet.plot(X, dep, unif.margin = FALSE, cex=0.5, ...)
```

Arguments

X	a matrix with two columns.
dep	depth of BET.
unif.margin	logicals. If TRUE the marginal distribution is known to be Uniform[0,1]. Default is FALSE, and empirical cdf transformation will be applied to each marginal distribution.
cex	number indicating the amount by which plotting text and symbols should be scaled relative to the default.
...	graphical parameters to plot

Examples

```
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
bet.plot(cbind(X1, X2), 3)
```

BET_package *Binary Expansion Testing*

Description

The BET package provides functions for nonparametric detection of nonuniformity and dependence with Binary Expansion Testing (BET).

BET functions

MaxBET symm get.signs cell.counts bet.plot MaxBETs BEAST BEAST.null.simu

Reference(s)

Kai Zhang (2019) BET on Independence, Journal of the American Statistical Association, 114:528, 1620-1637, doi: 10.1080/01621459.2018.1537921, Kai Zhang, Zhigen Zhao, and Wen Zhou (2021). BEAUTY Powered BEAST, <arXiv:2103.00674> and Zhigen Zhao, Michael Baiocchi, Kai Zhang. SorBET: A Fast and Powerful Algorithm to Test Dependence of Variables.

cell.counts *Counts the amount of points in each cell after binary expansion.*

Description

`cell.counts` returns the amount of data points in each cell getting from binary expansion.

Usage

```
cell.counts(X, dep, unif.margin = FALSE)
```

Arguments

`X`	a matrix to be tested.
`dep`	depth of the marginal binary expansions.
`unif.margin`	logicals. If `TRUE` the data has been uniformed based on empirical cumulative distribution function. Default to be `FALSE` and the function uniforms the data.

Value

The result is a dataframe with 2 rows and $2^(p * dep)$ columns, where p is the number of columns of `X`. The first column is the binary index, the second column is the amount of data points.

Examples

```
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
cell.counts(cbind(X1, X2), 3)
```

get.signs *Signs of Colors of all Points for all Interactions*

Description

`get.signs` returns all the signs of colors for each point under all interactions up to depth `d` in marginal binary expansions for the tests `BET` and `BETs`.

Usage

```
get.signs(X, dep, unif.margin = FALSE)
```

Arguments

X	a matrix to be tested.
dep	depth of the marginal binary expansions.
unif.margin	logicals. If TRUE the data has been uniformed based on empirical cumulative distribution function. Default to be FALSE and the function uniforms the data.

Value

The result is a dataframe with n rows and $2^(p * dep)$ columns, where p is the number of columns of X and n is the number of rows of X. The values of 1 or -1 stand for the sign of color, while the marginal interactions return 0.

Examples

```
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
get.signs(cbind(X1, X2), 3)
```

MaxBET *Binary Expansion Testing at a Certain Depth*

Description

MaxBET stands for Binary Expansion Testing. It is used for nonparametric detection of nonuniformity or dependence. It can be used to test whether a column vector is [0, 1]-uniformly distributed. It can also be used to detect dependence between columns of a matrix X, if X has more than one column.

Usage

```
MaxBET(
  X,
  dep,
  unif.margin = FALSE,
  asymptotic = TRUE,
  plot = FALSE,
  index = list(c(1:ncol(X)))
)
```

Arguments

X	a matrix to be tested. When X has only one column, BET will test whether X is [0, 1]-uniformly distributed (an error will be given if data is out of range [0, 1]). When X has two or more columns, BET tests the independence among those column vectors.

`dep`	depth of the binary expansion for the `BET`.
`unif.margin`	logicals. If `TRUE` the marginal distribution is known to be Uniform[0,1]. Default is `FALSE`, and empirical cdf transformation will be applied to each marginal distribution.
`asymptotic`	logicals. If `TRUE` the p-value is computed by asymptotic distribution. Default to be `TRUE`. Ignored if `X` has three or more columns.
`plot`	logicals. If `TRUE`, make the plot of cross interaction of the strongest asymmetry. Default to be `FALSE`. This option only works for X with two columns.
`index`	a list of indices. If provided, test the independence among two or more groups of variables. For example, `index = list(c(1,2), c(3))` refers to test the independence between (X_1, X_2) and X_3. Default to be `list(c(1:p))`, where `p = ncol(X)`, then test data uniformity.

Details

`MaxBET` tests the independence or uniformity by considering the maximal magnitude of the symmetry statistics in the $sigma$-field generated from marginal binary expansions at the depth `d`.

Value

`Interaction`	a dataframe with p columns, where p is the number of columns of `X`. It displays the interactions where the extreme symmetry statistics happens. For each column in `X`, we use a binary index to indicate binary variables involved in the extreme symmetry statistic.
`Extreme.Asymmetry`	the extreme asymmetry statistics.
`p.value.bonf`	p-value of the test with Bonferroni adjustment.
`z.statistic`	normal approximation of the test statistic.

Examples

```
##test mutual independence
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
MaxBET(cbind(X1, X2), 3, asymptotic = FALSE, index = list(1,2))

##test independence between (x1, x2) and y
x1 = runif(128)
x2 = runif(128)
y = sin(4*pi*(x1 + x2)) + 0.4*rnorm(128)
MaxBET(cbind(x1, x2, y), 3, index = list(c(1,2), c(3)))

##test uniformity
x1 = rbeta(128, 2, 4)
x2 = rbeta(128, 2, 4)
x3 = rbeta(128, 2, 4)
MaxBET(cbind(x1, x2, x3), 3)
```

MaxBETs *Binary Expansion Testing up to a Certain Depth*

Description

MaxBETs is used for nonparametric dependence detection. Extended from BET, for a chosen maximal depth d.max, MaxBETs does a sequential test up to d.max and avoids overlapping symmetry statistics in different depths, for all $2 \leq d \leq d.max$. The adjustment is done by multiplying the number of interactions which are in the σ-field generated by marginal binary expansions at depth d but not in that at depth $d-1$.

Usage

```
MaxBETs(
  X,
  d.max = 4,
  unif.margin = FALSE,
  asymptotic = TRUE,
  plot = FALSE,
  index = list(c(1:ncol(X)))
)
```

Arguments

X	a matrix to be tested. When X has only one column, BETs will test whether X is [0, 1]-uniformly distributed (an error will be given if data is out of range [0, 1]). When X has two or more columns, BETs tests the independence among those column vectors.
d.max	the maximal depth of the binary expansion for BETs.
unif.margin	logicals. If TRUE the marginal distribution is known to be Uniform[0,1]. Default is FALSE, and empirical cdf transformation will be applied to each marginal distribution.
asymptotic	logicals. If TRUE the p-value is computed by asymptotic distribution. Default to be TRUE. Ignored if X has three or more columns.
plot	logicals. If TRUE, make the plot of cross interaction of the strongest asymmetry. Default to be FALSE. This option only works for X with two columns.
index	a list of indices. If provided, test the independence among two or more groups of variables, for example, (X_1, X_2) and X_3.

Value

bet.s.pvalue.bonf	the overall p-value on the test.
bet.s.index	the interaction that the p-value is minimal.
bet.s.zstatistic	normal approximation of the test statistic.

Examples

```
##test mutual independence
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
MaxBETs(cbind(X1, X2), 3, asymptotic = FALSE, index = list(1,2))

##test independence between (x1, x2) and y
x1 = runif(128)
x2 = runif(128)
y = sin(4*pi*(x1 + x2)) + 0.4*rnorm(128)
MaxBETs(cbind(x1, x2, y), 3, index = list(c(1,2), c(3)))

##test uniformity
x1 = rbeta(128, 2, 4)
x2 = rbeta(128, 2, 4)
x3 = rbeta(128, 2, 4)
MaxBETs(cbind(x1, x2, x3), 3)
```

---

star *Coordinates of Brightest Stars in the Night Sky*

Description

This data set collects the galactic coordinates of the 256 brightest stars in the night sky (Perryman et al. 1997). We consider the longitude (x) and sine latitude (y) here.

Usage

```
data(star)
```

Format

An object of class `data.frame` with 256 rows and 2 columns.

Examples

```
data(star)
MaxBETs(cbind(star$x.raw, star$y.raw), asymptotic = FALSE, plot = TRUE, index = list(1,2))
```

symm *Symmetry Statistics for all Interactions*

Description

symm returns all the symmetry statistics up to depth d in marginal binary expansions for the tests BET and BETs.

Usage

```
symm(
  X,
  dep,
  unif.margin = FALSE,
  print.sample.size = TRUE
)
```

Arguments

X	a matrix to be tested.
dep	depth of the marginal binary expansions.
unif.margin	logicals. If TRUE the data has been uniformed based on empirical cumulative distribution function. Default to be FALSE and the function uniforms the data.
print.sample.size	logicals. If TRUE print the sample size. Default to be TRUE.

Value

The result is a dataframe with $(p + 2)$ columns, where p is the number of columns of X. The first column gives the binary index for all variables, the next p columns displays all the interactions of respective variables, the last column of Statistics gives the respective symmetry statistic.

Examples

```
v <- runif(128, -pi, pi)
X1 <- cos(v) + 2.5 * rnorm(128, 0, 1/20)
X2 <- sin(v) + 2.5 * rnorm(128, 0, 1/20)
symm(cbind(X1, X2), 3)
```

Index