Package 'Ball'

July 21, 2025

Type Package

Title Statistical Inference and Sure Independence Screening via Ball Statistics

Version 1.3.13

Date 2023-02-12

Maintainer Jin Zhu <zhuj37@mail2.sysu.edu.cn>

Description Hypothesis tests and sure independence screening (SIS) procedure based on ball statistics, including ball divergence <doi:10.1214/17-AOS1579>, ball covariance <doi:10.1080/01621459.2018.1543600>, and ball correlation <doi:10.1080/01621459.2018.1462709>, are developed to analyze complex data in metric spaces, e.g, shape, directional, compositional and symmetric positive definite matrix data. The ball divergence and ball covariance based distribution-free tests are implemented to detecting distribution difference and association in metric spaces <doi:10.18637/jss.v097.i06>. Furthermore, several generic non-parametric feature selection procedures based on ball correlation, BCor-SIS and all of its variants, are implemented to tackle the challenge in the context of ultra high dimensional data. A fast implementation for large-scale multiple K-sample testing with ball divergence <doi:10.1002/gepi.22423> is supported, which is particularly helpful for genome-wide association study.

License GPL-3

RoxygenNote 7.1.2

Depends R (>= 2.10)

Imports utils, gam, survival, mvtnorm

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Encoding UTF-8

LazyData true

URL https://mamba413.github.io/Ball/, https://github.com/Mamba413/Ball

BugReports https://github.com/Mamba413/Ball/issues

Author Jin Zhu [aut, cre] (ORCID: <https://orcid.org/0000-0001-8550-5822>),
Wenliang Pan [aut],
Yuan Tian [aut],
Weinan Xiao [aut],
Chengfeng Liu [aut],
Ruihuang Liu [aut],
Yue Hu [aut],
Hongtu Zhu [aut],
Heping Zhang [aut],
Xueqin Wang [aut] (ORCID: <https://orcid.org/0000-0001-5205-9950>)

Repository CRAN

Date/Publication 2023-02-12 10:00:02 UTC

Contents

ArcticLake 2
bcor 3
bcorsis 5
bcov.test 9
bd 13
bd.gwas.test 16
bd.test 18
bdvmf 22
genlung 23
macaques 24
meteorology 24
nhdist 25

Index **27**

`ArcticLake` *Arctic lake sediment samples of different water depth*

Description

Sand, silt and clay compositions of 39 sediment samples of different water depth in an Arctic lake.

Format

`ArcticLake$depth`: water depth (in meters).

`ArcticLake$x`: compositions of three covariates: sand, silt, and clay.

Details

Sand, silt and clay compositions of 39 sediment samples at different water depth (in meters) in an Arctic lake. The additional feature is a concomitant variable or covariate, water depth, which may account for some of the variation in the compositions. In statistical terminology, we have a multivariate regression problem with sediment composition as predictors and water depth as a response. All row percentage sums to 100, except for rounding errors.

Note

Courtesy of J. Aitchison

Source

Aitchison: CODA microcomputer statistical package, 1986, the file name ARCTIC.DAT, here included under the GNU Public Library Licence Version 2 or newer.

References

Aitchison: The Statistical Analysis of Compositional Data, 1986, Data 5, pp5.

---

bcor *Ball Covariance and Correlation Statistics*

Description

Computes Ball Covariance and Ball Correlation statistics, which are generic dependence measures in Banach spaces.

Usage

```
bcor(x, y, distance = FALSE, weight = FALSE)

bcov(x, y, distance = FALSE, weight = FALSE)
```

Arguments

x	a numeric vector, matrix, data.frame, or a list containing at least two numeric vectors, matrices, or data.frames.
y	a numeric vector, matrix, or data.frame.
distance	if distance = TRUE, the elements of x and y are considered as distance matrices.
weight	a logical or character string used to choose the weight form of Ball Covariance statistic.. If input is a character string, it must be one of "constant", "probability", or "chisquare". Any unambiguous substring can be given. If input is a logical value, it is equivalent to weight = "probability" if weight = TRUE while equivalent to weight = "constant" if weight = FALSE. Default: weight = FALSE.

Details

The sample sizes of the two variables must agree, and samples must not contain missing and infinite values. If we set `distance = TRUE`, arguments `x`, `y` can be a `dist` object or a symmetric numeric matrix recording distance between samples; otherwise, these arguments are treated as data.

`bcov` and `bcor` compute Ball Covariance and Ball Correlation statistics.

Ball Covariance statistics is a generic dependence measure in Banach spaces. It enjoys the following properties:

- It is nonnegative and it is equal to zero if and only if variables are unassociated;
- It is highly robust;
- It is distribution-free and model-free;
- it is interesting that the HHG is a special case of Ball Covariance statistics.

Ball correlation statistics, a normalized version of Ball Covariance statistics, generalizes Pearson correlation in two fundamental ways:

- It is well-defined for random variables in arbitrary dimension in Banach spaces
- BCor is equal to zero implies random variables are unassociated.

The definitions of the Ball Covariance and Ball Correlation statistics between two random variables are as follows. Suppose, we are given pairs of independent observations $\{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i and y_i can be of any dimension and the dimensionality of x_i and y_i need not be the same. Then, we define sample version Ball Covariance as:

$$\mathbf{BCov}^2_{\omega,n}(X, Y) = \frac{1}{n^2}\sum_{i,j=1}^{n}(\Delta^{X,Y}_{ij,n} - \Delta^{X}_{ij,n}\Delta^{Y}_{ij,n})^2$$

where:

$$\Delta^{X,Y}_{ij,n} = \frac{1}{n}\sum_{k=1}^{n}\delta^{X}_{ij,k}\delta^{Y}_{ij,k}, \Delta^{X}_{ij,n} = \frac{1}{n}\sum_{k=1}^{n}\delta^{X}_{ij,k}, \Delta^{Y}_{ij,n} = \frac{1}{n}\sum_{k=1}^{n}\delta^{Y}_{ij,k}$$

$$\delta^{X}_{ij,k} = I(x_k \in \bar{B}(x_i, \rho(x_i, x_j))), \delta^{Y}_{ij,k} = I(y_k \in \bar{B}(y_i, \rho(y_i, y_j)))$$

Among them, $\bar{B}(x_i, \rho(x_i, x_j))$ is a closed ball with center x_i and radius $\rho(x_i, x_j)$. Similarly, we can define $\mathbf{BCov}^2_{\omega,n}(\mathbf{X}, \mathbf{X})$ and $\mathbf{BCov}^2_{\omega,n}(\mathbf{Y}, \mathbf{Y})$. We define Ball Correlation statistic as follows.

$$\mathbf{BCor}^2_{\omega,n}(\mathbf{X}, \mathbf{Y}) = \mathbf{BCov}^2_{\omega,n}(\mathbf{X}, \mathbf{Y})/\sqrt{\mathbf{BCov}^2_{\omega,n}(\mathbf{X}, \mathbf{X})\mathbf{BCov}^2_{\omega,n}(\mathbf{Y}, \mathbf{Y})}$$

We can extend $\mathbf{BCov}_{\omega,n}$ to measure the mutual independence between K random variables:

$$\frac{1}{n^2}\sum_{i,j=1}^{n}\left[(\Delta^{X_1,\ldots,X_K}_{ij,n} - \prod_{k=1}^{K}\Delta^{X_k}_{ij,n})^2\prod_{k=1}^{K}\hat{\omega}_k(X_{ki}, X_{kj})\right]$$

where $X_k(k = 1, \ldots, K)$ are random variables and X_{ki} is the i-th observations of X_k.

See `bcov.test` for a test of independence based on the Ball Covariance statistic.

Value

bcor	Ball Correlation statistic.
bcov	Ball Covariance statistic.

References

Wenliang Pan, Xueqin Wang, Heping Zhang, Hongtu Zhu & Jin Zhu (2019) Ball Covariance: A Generic Measure of Dependence in Banach Space, Journal of the American Statistical Association, DOI: 10.1080/01621459.2018.1543600

Wenliang Pan, Xueqin Wang, Weinan Xiao & Hongtu Zhu (2018) A Generic Sure Independence Screening Procedure, Journal of the American Statistical Association, DOI: 10.1080/01621459.2018.1462709

Jin Zhu, Wenliang Pan, Wei Zheng, and Xueqin Wang (2021). Ball: An R Package for Detecting Distribution Difference and Association in Metric Spaces, Journal of Statistical Software, Vol.97(6), doi: 10.18637/jss.v097.i06.

See Also

bcov.test, bcorsis

Examples

```
############# Ball Correlation #############
num <- 50
x <- 1:num
y <- 1:num
bcor(x, y)
bcor(x, y, weight = "prob")
bcor(x, y, weight = "chisq")
############# Ball Covariance #############
num <- 50
x <- rnorm(num)
y <- rnorm(num)
bcov(x, y)
bcov(x, y, weight = "prob")
bcov(x, y, weight = "chisq")
```

---

bcorsis *Ball Correlation based Sure Independence Screening (BCor-SIS)*

Description

Generic non-parametric sure independence screening (SIS) procedure based on Ball Correlation. Ball correlation is a generic measure of dependence in Banach spaces.

Usage

```
bcorsis(
  x,
  y,
  d = "small",
  weight = c("constant", "probability", "chisquare"),
  method = "standard",
  distance = FALSE,
  category = FALSE,
  parms = list(d1 = 5, d2 = 5, df = 3),
  num.threads = 0
)
```

Arguments

x	a numeric matrix or data.frame included n rows and p columns. Each row is an observation vector and each column corresponding to a explanatory variable, generally $p >> n$.
y	a numeric vector, matrix, or data.frame.
d	the hard cutoff rule suggests selecting d variables. Setting d = "large" or d = "small" means n - 1 or floor(n/log(n)) variables are selected. If d is a integer, d variables are selected. Default: d = "small".
weight	a logical or character string used to choose the weight form of Ball Covariance statistic.. If input is a character string, it must be one of "constant", "probability", or "chisquare". Any unambiguous substring can be given. If input is a logical value, it is equivalent to weight = "probability" if weight = TRUE while equivalent to weight = "constant" if weight = FALSE. Default: weight = FALSE.
method	specific method for the BCor-SIS procedure. It must be one of "standard", "lm", "gam", "interaction", or "survival". Setting method = "standard" means performing standard SIS procedure while the options "lm" and "gam" mean carrying out iterative SIS procedure with ordinary linear regression and generalized additive models, respectively. The options "interaction" and "survival" are designed for detecting variables with potential linear interaction and associated with left censored responses, respectively. Any unambiguous substring can be given. Default: method = "standard".
distance	if distance = TRUE, y will be considered as a distance matrix. Arguments only available when method = "standard" and method = "interaction". Default: distance = FALSE.
category	a logical value or integer vector indicating columns to be selected as categorical variables. If category is an integer vector, the positive/negative integers select/discard the corresponding columns; If category is a logical value, category = TRUE select all columns, category = FALSE select none column. Default: category = FALSE.
parms	parameters list only available when method = "lm" or "gam". It contains three parameters: d1, d2, and df. d1 is the number of initially selected variables, d2

is the number of variables added in each iteration. `df` is a degree freedom of basis in generalized additive models playing a role only when `method = "gam"`. Default: `parms = list(d1 = 5, d2 = 5, df = 3)`.

`num.threads` number of threads. If `num.threads = 0`, then all of available cores will be used. Default `num.threads = 0`.

Details

`bcorsis` performs a model-free generic sure independence screening procedure, BCor-SIS, to pick out variables from `x` which are potentially associated with `y`. BCor-SIS relies on Ball correlation, a universal dependence measure in Banach spaces. Ball correlation (BCor) ranges from 0 to 1. A larger BCor implies they are likely to be associated while Bcor is equal to 0 implies they are unassociated. (See `bcor` for details.) Consequently, BCor-SIS pick out variables with larger Bcor values with `y`.

Theory and numerical result indicate that BCor-SIS has following advantages:

- BCor-SIS can retain the efficient variables even when the dimensionality (i.e., `ncol(x)`) is an exponential order of the sample size (i.e., `exp(nrow(x))`);
- It is distribution-free and model-free;
- It is very robust;
- It is works well for complex data, such as shape and survival data;

If `x` is a matrix, the sample sizes of `x` and `y` must agree. If `x` is a `list` object, each element of this `list` must with the same sample size. `x` and `y` must not contain missing or infinite values.

When `method = "survival"`, the matrix or data.frame pass to `y` must have exactly two columns, where the first column is event (failure) time while the second column is a dichotomous censored status.

Value

`ix` the indices vector corresponding to variables selected by BCor-SIS.

`method` the method used.

`weight` the weight used.

`complete.info` a `list` mainly containing a $p \times 3$ matrix, where each row is a variable and each column is a weight Ball Correlation statistic. If `method = "gam"` or `method = "lm"`, `complete.info` is an empty list.

Note

`bcorsis` simultaneously computing Ball Correlation statistics with `"constant"`, `"probability"`, and `"chisquare"` weights. Users can get other Ball Correlation statistics with different weight in the `complete.info` element of output. We give a quick example below to illustrate.

Author(s)

Wenliang Pan, Weinan Xiao, Xueqin Wang, Hongtu Zhu, Jin Zhu

References

Wenliang Pan, Xueqin Wang, Weinan Xiao & Hongtu Zhu (2018) A Generic Sure Independence Screening Procedure, Journal of the American Statistical Association, DOI: 10.1080/01621459.2018.1462709

See Also

bcor

Examples

```
## Not run:

############### Quick Start for bcorsis function ###############
set.seed(1)
n <- 150
p <- 3000
x <- matrix(rnorm(n * p), nrow = n)
eps <- rnorm(n)
y <- 3 * x[, 1] + 5 * (x[, 3])^2 + eps
res <- bcorsis(y = y, x = x)
head(res[["ix"]])
head(res[["complete.info"]][["statistic"]])

############### BCor-SIS: Censored Data Example ###############
data("genlung")
result <- bcorsis(x = genlung[["covariate"]], y = genlung[["survival"]],
                  method = "survival")
index <- result[["ix"]]
top_gene <- colnames(genlung[["covariate"]])[index]
head(top_gene, n = 1)


############### BCor-SIS: Interaction Pursuing ###############
set.seed(1)
n <- 150
p <- 3000
x <- matrix(rnorm(n * p), nrow = n)
eps <- rnorm(n)
y <- 3 * x[, 1] * x[, 5] * x[, 10] + eps
res <- bcorsis(y = y, x = x, method = "interaction")
head(res[["ix"]])

############### BCor-SIS: Iterative Method ###############
library(mvtnorm)
set.seed(1)
n <- 150
p <- 3000
sigma_mat <- matrix(0.5, nrow = p, ncol = p)
diag(sigma_mat) <- 1
x <- rmvnorm(n = n, sigma = sigma_mat)
eps <- rnorm(n)
rm(sigma_mat); gc(reset = TRUE)
```

```
y <- 3 * (x[, 1])^2 + 5 * (x[, 2])^2 + 5 * x[, 8] - 8 * x[, 16] + eps
res <- bcorsis(y = y, x = x, method = "lm", d = 15)
res <- bcorsis(y = y, x = x, method = "gam", d = 15)
res[["ix"]]

############### Weighted BCor-SIS: Probability weight ###############
set.seed(1)
n <- 150
p <- 3000
x <- matrix(rnorm(n * p), nrow = n)
eps <- rnorm(n)
y <- 3 * x[, 1] + 5 * (x[, 3])^2 + eps
res <- bcorsis(y = y, x = x, weight = "prob")
head(res[["ix"]])
# Alternative, chisq weight:
res <- bcorsis(y = y, x = x, weight = "chisq")
head(res[["ix"]])

############### BCor-SIS: GWAS data ###############
set.seed(1)
n <- 150
p <- 3000
x <- sapply(1:p, function(i) {
  sample(0:2, size = n, replace = TRUE)
})
eps <- rnorm(n)
y <- 6 * x[, 1] - 7 * x[, 2] + 5 * x[, 3] + eps
res <- bcorsis(x = x, y = y, category = TRUE)
head(res[["ix"]])
head(res[["complete.info"]][["statistic"]])

x <- cbind(matrix(rnorm(n * 2), ncol = 2), x)
# remove the first two columns:
res <- bcorsis(x = x, y = y, category = c(-1, -2))
head(res[["ix"]])

x <- cbind(x[, 3:5], matrix(rnorm(n * p), ncol = p))
res <- bcorsis(x = x, y = y, category = 1:3)
head(res[["ix"]], n = 10)

## End(Not run)
```

`bcov.test` *Ball Covariance Test*

Description

Ball Covariance test of independence. Ball covariance are generic dependence measures in Banach spaces.

Usage

```
bcov.test(x, ...)

## Default S3 method:
bcov.test(
  x,
  y = NULL,
  num.permutations = 99,
  method = c("permutation", "limit"),
  distance = FALSE,
  weight = FALSE,
  seed = 1,
  num.threads = 0,
  ...
)

## S3 method for class 'formula'
bcov.test(formula, data, subset, na.action, ...)
```

Arguments

x	a numeric vector, matrix, data.frame, or a list containing at least two numeric vectors, matrices, or data.frames.
...	further arguments to be passed to or from methods.
y	a numeric vector, matrix, or data.frame.
num.permutations	the number of permutation replications. When num.permutations = 0, the function just returns the Ball Covariance statistic. Default: num.permutations = 99.
method	if method = "permutation", a permutation procedure is carried out to compute the p-value; if method = "limit", an approximate null distribution is used when weight = "constant". Any unambiguous substring can be given. Default method = "permutation".
distance	if distance = TRUE, the elements of x and y are considered as distance matrices.
weight	a logical or character string used to choose the weight form of Ball Covariance statistic.. If input is a character string, it must be one of "constant", "probability", or "chisquare". Any unambiguous substring can be given. If input is a logical value, it is equivalent to weight = "probability" if weight = TRUE while equivalent to weight = "constant" if weight = FALSE. Default: weight = FALSE.
seed	the random seed. Default seed = 1.
num.threads	number of threads. If num.threads = 0, then all of available cores will be used. Default num.threads = 0.
formula	a formula of the form ~ u + v, where each of u and v are numeric variables giving the data values for one sample. The samples must be of the same length.

data	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

bcov.test is non-parametric tests of independence in Banach spaces. It can detect the dependence between two random objects (variables) and the mutual dependence among at least three random objects (variables).

If two samples are pass to arguments x and y, the sample sizes (i.e. number of rows or length of the vector) of the two variables must agree. If a list object is passed to x, this list must contain at least two numeric vectors, matrices, or data.frames, and each element of this list must with the same sample size. Moreover, data pass to x or y must not contain missing or infinite values. If distance = TRUE, x is considered as a distance matrix or a list containing distance matrices, and y is considered as a distance matrix; otherwise, these arguments are treated as data.

bcov.test utilizes the Ball Covariance statistics (see bcov) to measure dependence and derives a p-value via replicating the random permutation num.permutations times.

See Pan et al 2018 for theoretical properties of the test, including statistical consistency.

Value

If num.permutations > 0, bcov.test returns a htest class object containing the following components:

statistic	Ball Covariance statistic.
p.value	the p-value for the test.
replicates	permutation replications of the test statistic.
size	sample size.
complete.info	a list mainly containing two vectors, the first vector is the Ball Covariance statistics with different weights, the second is the p-values of weighted Ball Covariance tests.
alternative	a character string describing the alternative hypothesis.
method	a character string indicating what type of test was performed.
data.name	description of data.

If num.permutations = 0, bcov.test returns a statistic value.

Note

Actually, bcov.test simultaneously computing Ball Covariance statistics with "constant", "probability", and "chisquare" weights. Users can get other Ball Covariance statistics with different weight and their corresponding p-values in the complete.info element of output. We give a quick example below to illustrate.

Author(s)

Wenliang Pan, Xueqin Wang, Heping Zhang, Hongtu Zhu, Jin Zhu

References

Wenliang Pan, Xueqin Wang, Heping Zhang, Hongtu Zhu & Jin Zhu (2019) Ball Covariance: A Generic Measure of Dependence in Banach Space, Journal of the American Statistical Association, DOI: 10.1080/01621459.2018.1543600

Jin Zhu, Wenliang Pan, Wei Zheng, and Xueqin Wang (2021). Ball: An R Package for Detecting Distribution Difference and Association in Metric Spaces, Journal of Statistical Software, Vol.97(6), doi: 10.18637/jss.v097.i06.

See Also

bcov, bcor

Examples

```
set.seed(1)

################## Quick Start #################
noise <- runif(50, min = -0.3, max = 0.3)
x <- runif(50, 0, 4*pi)
y <- cos(x) + noise
# plot(x, y)
res <- bcov.test(x, y)
res
## get all Ball Covariance statistics:
res[["complete.info"]][["statistic"]]
## get all test result:
res[["complete.info"]][["p.value"]]

################## Quick Start #################
x <- matrix(runif(50 * 2, -pi, pi), nrow = 50, ncol = 2)
noise <- runif(50, min = -0.1, max = 0.1)
y <- sin(x[,1] + x[,2]) + noise
bcov.test(x = x, y = y, weight = "prob")

################## Ball Covariance Test for Non-Hilbert Data #################
# load data:
data("ArcticLake")
# Distance matrix between y:
Dy <- nhdist(ArcticLake[["x"]], method = "compositional")
# Distance matrix between x:
Dx <- dist(ArcticLake[["depth"]])
# hypothesis test with BCov:
bcov.test(x = Dx, y = Dy, distance = TRUE)

################  Weighted Ball Covariance Test  #################
data("ArcticLake")
Dy <- nhdist(ArcticLake[["x"]], method = "compositional")
```

```
Dx <- dist(ArcticLake[["depth"]])
# hypothesis test with weighted BCov:
bcov.test(x = Dx, y = Dy, distance = TRUE, weight = "prob")

################# Mutual Independence Test #################
x <- rnorm(50)
y <- (x > 0) * x + rnorm(50)
z <- (x <= 0) * x + rnorm(50)
data_list <- list(x, y, z)
bcov.test(data_list)
data_list <- lapply(data_list, function(x) {
  as.matrix(dist(x))
})
bcov.test(data_list, distance = TRUE)
bcov.test(data_list, distance = FALSE, weight = "chi")

################# Mutual Independence Test for Meteorology data #################
data("meteorology")
bcov.test(meteorology)

################  Testing via approximate limit distribution  #################
## Not run:
set.seed(1)
n <- 2000
x <- rnorm(n)
y <- rnorm(n)
bcov.test(x, y, method = "limit")
bcov.test(x, y)

## End(Not run)

################  Formula interface  ################
## independence test:
bcov.test(~ CONT + INTG, data = USJudgeRatings)
## independence test with chisquare weight:
bcov.test(~ CONT + INTG, data = USJudgeRatings, weight = "chi")
## mutual independence test:
bcov.test(~ CONT + INTG + DMNR, data = USJudgeRatings)
```

---

bd *Ball Divergence statistic*

Description

Compute Ball Divergence statistic, which is a generic dispersion measure in Banach spaces.

Usage

```
bd(
  x,
```

```
  y = NULL,
  distance = FALSE,
  size = NULL,
  num.threads = 1,
  kbd.type = c("sum", "maxsum", "max")
)
```

Arguments

x	a numeric vector, matrix, data.frame, or a list containing at least two numeric vectors, matrices, or data.frames.
y	a numeric vector, matrix, data.frame.
distance	if distance = TRUE, the elements of x will be considered as a distance matrix. Default: distance = FALSE.
size	a vector recording sample size of each group.
num.threads	number of threads. If num.threads = 0, then all of available cores will be used. Default num.threads = 0.
kbd.type	a character string specifying the K-sample Ball Divergence test statistic, must be one of "sum", "summax", or "max". Any unambiguous substring can be given. Default kbd.type = "sum".

Details

Given the samples not containing missing values, bd returns Ball Divergence statistics. If we set distance = TRUE, arguments x, y can be a dist object or a symmetric numeric matrix recording distance between samples; otherwise, these arguments are treated as data.

Ball divergence statistic measure the distribution difference of two datasets in Banach spaces. The Ball divergence statistic is proven to be zero if and only if two datasets are identical.

The definition of the Ball Divergence statistics is as follows. Given two independent samples $\{x_1, \ldots, x_n\}$ with the associated probability measure μ and $\{y_1, \ldots, y_m\}$ with ν, where the observations in each sample are *i.i.d.* Let $\delta(x, y, z) = I(z \in \bar{B}(x, \rho(x, y)))$, where $\delta(x, y, z)$ indicates whether z is located in the closed ball $\bar{B}(x, \rho(x, y))$ with center x and radius $\rho(x, y)$. We denote:

$$A_{ij}^X = \frac{1}{n}\sum_{u=1}^{n}\delta(X_i, X_j, X_u), \quad A_{ij}^Y = \frac{1}{m}\sum_{v=1}^{m}\delta(X_i, X_j, Y_v),$$

$$C_{kl}^X = \frac{1}{n}\sum_{u=1}^{n}\delta(Y_k, Y_l, X_u), \quad C_{kl}^Y = \frac{1}{m}\sum_{v=1}^{m}\delta(Y_k, Y_l, Y_v).$$

A_{ij}^X represents the proportion of samples $\{x_1, \ldots, x_n\}$ located in the ball $\bar{B}(X_i, \rho(X_i, X_j))$ and A_{ij}^Y represents the proportion of samples $\{y_1, \ldots, y_m\}$ located in the ball $\bar{B}(X_i, \rho(X_i, X_j))$. Meanwhile, C_{kl}^X and C_{kl}^Y represent the corresponding proportions located in the ball $\bar{B}(Y_k, \rho(Y_k, Y_l))$. The Ball Divergence statistic is defined as:

$$D_{n,m} = A_{n,m} + C_{n,m}$$

Ball Divergence can be generalized to the K-sample test problem. Suppose we have K group samples, each group include n_k samples. The definition of K-sample Ball Divergence statistic

could be to directly sum up the two-sample Ball Divergence statistics of all sample pairs (`kbd.type = "sum"`)

$$\sum_{1\leq k<l\leq K} D_{n_k,n_l},$$

or to find one sample with the largest difference to the others (`kbd.type = "maxsum"`)

$$\max_t \sum_{s=1,s\neq t}^{K} D_{n_s,n_t},$$

to aggregate the $K-1$ most significant different two-sample Ball Divergence statistics (`kbd.type = "max"`)

$$\sum_{k=1}^{K-1} D_{(k)},$$

where $D_{(1)},\ldots,D_{(K-1)}$ are the largest $K-1$ two-sample Ball Divergence statistics among $\{D_{n_s,n_t}|1\leq s<t\leq K\}$. When $K=2$, the three types of Ball Divergence statistics degenerate into two-sample Ball Divergence statistic.

See `bd.test` for a test of distribution equality based on the Ball Divergence.

Value

`bd`	Ball Divergence statistic

Author(s)

Wenliang Pan, Yuan Tian, Xueqin Wang, Heping Zhang

References

Wenliang Pan, Yuan Tian, Xueqin Wang, Heping Zhang. Ball Divergence: Nonparametric two sample test. Ann. Statist. 46 (2018), no. 3, 1109–1137. doi:10.1214/17-AOS1579. https://projecteuclid.org/euclid.aos/15253130

See Also

`bd.test`

Examples

```
############# Ball Divergence #############
x <- rnorm(50)
y <- rnorm(50)
bd(x, y)
```

bd.gwas.test *Fast K-sample Ball Divergence Test for GWAS Data*

Description

Fast K-sample Ball Divergence Test for GWAS Data

Usage

```
bd.gwas.test(
  x,
  snp,
  screening.method = c("permute", "spectrum"),
  refine = TRUE,
  num.permutations,
  distance = FALSE,
  alpha,
  screening.result = NULL,
  verbose = TRUE,
  seed = 1,
  num.threads = 0,
  ...
)
```

Arguments

x	a numeric vector, matrix, data.frame, dist object.
snp	a numeric matrix recording the values of single nucleotide polymorphism (SNP). Each column must be an integer vector.
screening.method	if screening.method = "spectrum", the spectrum method is applied to screening the candidate SNPs, or otherwise, the permutation method is applied. Default: screening.method = "permute".
refine	a logical value. If refine = TRUE, a p-values refining process is applied to the SNPs which passes the pre-screening process. Default: refine = TRUE (At present, refine = FALSE is not available).
num.permutations	the number of permutation replications. When num.permutations = 0, the function just returns the Ball Divergence statistic. Default: num.permutations = 100 * ncol(snp)
distance	if distance = TRUE, the elements of x will be considered as a distance matrix. Default: distance = FALSE.
alpha	the significance level. Default: 0.05 / ncol(snp).

`screening.result`
: A object return by `bd.gwas.test` that preserving the pre-screening result. It works only if the pre-screening is available. Default: `screening.result = NULL`.

`verbose`
: Show computation status and estimated runtimes. Default: `verbose = FALSE`.

`seed`
: the random seed. Default `seed = 1`.

`num.threads`
: number of threads. If `num.threads = 0`, then all of available cores will be used. Default `num.threads = 0`.

`...`
: further arguments to be passed to or from methods.

Value

bd.gwas.test returns a list containing the following components:

`statistic`
: ball divergence statistics vector.

`permuted.statistic`
: a data.frame containing permuted ball divergence statistic for pre-screening SNPs. If `refine = FALSE`, it takes value `NULL`.

`eigenvalue`
: the eigenvalue of spectrum decomposition. If `refine = TRUE`, it takes value `NULL`.

`p.value`
: the p-values of ball divergence test.

`refined.snp`
: the SNPs have been refined.

`refined.p.value`
: the refined p-value of significant snp.

`refined.permuted.statistic`
: a data.frame containing permuted ball divergence statistics for refining p-values.

`screening.result`
: a list containing the result of screening.

Author(s)

Jin Zhu

References

Yue Hu, Haizhu Tan, Cai Li, and Heping Zhang. (2021). Identifying genetic risk variants associated with brain volumetric phenotypes via K-sample Ball Divergence method. Genetic Epidemiology, 1–11. https://doi.org/10.1002/gepi.22423

See Also

`bd`, `bd.test`

Examples

```
library(Ball)
set.seed(1234)
num <- 200
snp_num <- 500
p <- 5
x <- matrix(rnorm(num * p), nrow = num)
snp <- sapply(1:snp_num, function(i) {
  sample(0:2, size = num, replace = TRUE)
})
snp1 <- sapply(1:snp_num, function(i) {
  sample(1:2, size = num, replace = TRUE)
})
snp <- cbind(snp, snp1)
res <- Ball::bd.gwas.test(x = x, snp = snp)
mean(res[["p.value"]] < 0.05)
mean(res[["p.value"]] < 0.005)

## only return the test statistics;
res <- Ball::bd.gwas.test(x = x, snp = snp, num.permutation = 0)

## save pre-screening process results:
x <- matrix(rnorm(num * p), nrow = num)
snp <- sapply(1:snp_num, function(i) {
  sample(0:2, size = num, replace = TRUE, prob = c(1/2, 1/4, 1/4))
})
snp_screening <- Ball::bd.gwas.test(x = x, snp = snp,
                                    alpha = 5*10^-4,
                                    num.permutations = 19999)
mean(res[["p.value"]] < 0.05)
mean(res[["p.value"]] < 0.005)
mean(res[["p.value"]] < 0.0005)
## refine p-value according to the pre-screening process result:
res <- Ball::bd.gwas.test(x = x, snp = snp, alpha = 5*10^-4,
                          num.permutations = 19999,
                          screening.result = snp_screening[["screening.result"]])
```

---

bd.test *Ball Divergence based Equality of Distributions Test*

Description

Performs the nonparametric two-sample or K-sample Ball Divergence test for equality of multivariate distributions

Usage

```
bd.test(x, ...)
```

```
## Default S3 method:
bd.test(
  x,
  y = NULL,
  num.permutations = 99,
  method = c("permutation", "limit"),
  distance = FALSE,
  size = NULL,
  seed = 1,
  num.threads = 0,
  kbd.type = c("sum", "maxsum", "max"),
  weight = c("constant", "variance"),
  ...
)

## S3 method for class 'formula'
bd.test(formula, data, subset, na.action, ...)
```

Arguments

x	a numeric vector, matrix, data.frame, or a list containing at least two numeric vectors, matrices, or data.frames.
...	further arguments to be passed to or from methods.
y	a numeric vector, matrix, data.frame.
num.permutations	the number of permutation replications. When num.permutations = 0, the function just returns the Ball Divergence statistic. Default: num.permutations = 99.
method	if method = "permutation", a permutation procedure is carried out to compute the p-value; if method = "limit", an approximate null distribution is used when weight = "constant". Any unambiguous substring can be given. Default method = "permutation".
distance	if distance = TRUE, the elements of x will be considered as a distance matrix. Default: distance = FALSE.
size	a vector recording sample size of each group.
seed	the random seed. Default seed = 1.
num.threads	number of threads. If num.threads = 0, then all of available cores will be used. Default num.threads = 0.
kbd.type	a character string specifying the K-sample Ball Divergence test statistic, must be one of "sum", "summax", or "max". Any unambiguous substring can be given. Default kbd.type = "sum".
weight	a character string specifying the weight form of Ball Divergence statistic. It must be one of "constant" or "variance". Any unambiguous substring can be given. Default: weight = "constant".
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.

data	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

bd.test is nonparametric test for the two-sample or K-sample problem. It can detect distribution difference between $K(K \geq 2)$ sample even though sample size are imbalanced. This test can cope well multivariate dataset or complex dataset.

If only x is given, the statistic is computed from the original pooled samples, stacked in matrix where each row is a multivariate observation, or from the distance matrix when distance = TRUE. The first sizes[1] rows of x are the first sample, the next sizes[2] rows of x are the second sample, etc. If x is a list, its elements are taken as the samples to be compared, and hence, this list must contain at least two numeric data vectors, matrices or data.frames.

bd.test utilizes the Ball Divergence statistics (see bd) to measure dispersion and derives a p-value via replicating the random permutation num.permutations times. The function simply returns the test statistic when num.permutations = 0.

The time complexity of bd.test is around $O(R \times n^2)$, where R = num.permutations and n is sample size.

Value

If num.permutations > 0, bd.test returns a htest class object containing the following components:

statistic	Ball Divergence statistic.
p.value	the p-value for the test.
replicates	permutation replications of the test statistic.
size	sample sizes.
complete.info	a list mainly containing two vectors, the first vector is the Ball Divergence statistics with different aggregation strategy and weight, the second vector is the p-values of tests.
alternative	a character string describing the alternative hypothesis.
method	a character string indicating what type of test was performed.
data.name	description of data.

If num.permutations = 0, bd.test returns a statistic value.

Note

Actually, bd.test simultaneously computing "sum", "summax", and "max" Ball Divergence statistics when $K \geq 3$. Users can get other Ball Divergence statistics and their corresponding p-values in the complete.info element of output. We give a quick example below to illustrate.

Author(s)

Wenliang Pan, Yuan Tian, Xueqin Wang, Heping Zhang, Jin Zhu

References

Wenliang Pan, Yuan Tian, Xueqin Wang, Heping Zhang. Ball Divergence: Nonparametric two sample test. Annals of Statistics. 46 (2018), no. 3, 1109–1137. doi:10.1214/17-AOS1579. https://projecteuclid.org/euclid.aos/15

Jin Zhu, Wenliang Pan, Wei Zheng, and Xueqin Wang (2021). Ball: An R Package for Detecting Distribution Difference and Association in Metric Spaces, Journal of Statistical Software, Vol.97(6), doi: 10.18637/jss.v097.i06.

See Also

bd

Examples

```
################## Quick Start #################
set.seed(1)
x <- rnorm(50)
y <- rnorm(50, mean = 1)
# plot(density(x))
# lines(density(y), col = "red")
bd.test(x = x, y = y)

################## Quick Start #################
x <- matrix(rnorm(100), nrow = 50, ncol = 2)
y <- matrix(rnorm(100, mean = 3), nrow = 50, ncol = 2)
# Hypothesis test with Standard Ball Divergence:
bd.test(x = x, y = y)

################## Simlated Non-Hilbert data #################
data("bdvmf")
## Not run:
library(scatterplot3d)
scatterplot3d(bdvmf[["x"]], color = bdvmf[["group"]],
              xlab = "X1", ylab = "X2", zlab = "X3")

## End(Not run)
# calculate geodesic distance between sample:
Dmat <- nhdist(bdvmf[["x"]], method = "geodesic")
# hypothesis test with BD :
bd.test(x = Dmat, size = c(150, 150), num.permutations = 99, distance = TRUE)

################## Non-Hilbert Real Data #################
# load data:
data("macaques")
# number of femala and male Macaca fascicularis:
table(macaques[["group"]])
# calculate Riemannian shape distance matrix:
Dmat <- nhdist(macaques[["x"]], method = "riemann")
```

```
# hypothesis test with BD:
bd.test(x = Dmat, num.permutations = 99, size = c(9, 9), distance = TRUE)

################  K-sample Test  #################
n <- 150
bd.test(rnorm(n), size = c(40, 50, 60))
# alternative input method:
x <- lapply(c(40, 50, 60), rnorm)
res <- bd.test(x)
res
## get all Ball Divergence statistics:
res[["complete.info"]][["statistic"]]
## get all test result:
res[["complete.info"]][["p.value"]]

################  Testing via approximate limit distribution  #################
## Not run:
set.seed(1)
n <- 1000
x <- rnorm(n)
y <- rnorm(n)
res <- bd.test(x, y, method = "limit")
bd.test(x, y)

## End(Not run)

################  Formula interface  ################
## Two-sample test
bd.test(extra ~ group, data = sleep)
## K-sample test
bd.test(Sepal.Width ~ Species, data = iris)
bd.test(Sepal.Width ~ Species, data = iris, kbd.type = "max")
```

---

bdvmf *Simulated von Mises-Fisher Data*

Description

Simulated random vectors following the von Mises-Fisher distribution with mean direction $\mu_x = (1, 0, 0)$ and $\mu_y = (1, 1, 1)$, and concentration parameter is $\kappa = 3$.

Format

bdvmf$x: A 300×3 numeric matrix containing simulated von Mises-Fisher data.

bdvmf$group: A group index vector.

Details

In directional statistics, the von Mises–Fisher distribution (named after Ronald Fisher and Richard von Mises), is a probability distribution on the $(p-1)$-dimensional sphere in R^p

The parameters μ, and κ, are called the mean direction and concentration parameter, respectively. The greater the value of κ, the higher the concentration of the distribution around the mean direction μ,. The distribution is unimodal for κ, and is uniform on the sphere for $\kappa = 0$.

References

Embleton, N. I. Fisher, T. Lewis, B. J. J. (1993). Statistical analysis of spherical data (1st pbk. ed.). Cambridge: Cambridge University Press. pp. 115–116. ISBN 0-521-45699-1.

`genlung` *Lung cancer genomic data*

Description

Publicly available lung cancer genomic data from the Chemores Cohort Study, containing the expression levels of mRNA, miRNA, artificial noise variables as well as clinical variables and response.

Format

`genlung$survival`: A data.frame containing $n = 123$ complete observations. The first column is disease-free survival time and the second column is censoring status.

`genlung$covariate`: A data.frame containing $p = 2000$ covariates.

Details

Tissue samples were analysed from a cohort of 123 patients, who underwent complete surgical resection at the Institut Mutualiste Montsouris (Paris, France) between 30 January 2002 and 26 June 2006. The studied outcome was the "Disease-Free Survival Time". Patients were followed until the first relapse occurred or administrative censoring. In this genomic dataset, the expression levels of Agilent miRNA probes ($p = 939$) were included from the $n = 123$ cohort samples. The miRNA data contains normalized expression levels. See below the paper by Lazar et al. (2013) and Array Express data repository for the complete description of the samples, tissue preparation, Agilent array technology, and data normalization. In addition to the genomic data, five clinical variables, also evaluated on the cohort samples, are included as continuous variable ('Age') and nominal variables ('Type','KRAS.status','EGFR.status','P53.status'). See Lazar et al. (2013) for more details. Moreover, we add 1056 standard gaussian variables which are independent with the censored response as noise covariates. This dataset represents a situation where the number of covariates dominates the number of complete observations or $p >> n$ case.

References

Lazar V. et al. (2013). Integrated molecular portrait of non-small cell lung cancers. BMC Medical Genomics 6:53-65.

macaques *Male and Female macaque data*

Description

Male and female macaque skull data. 7 landmarks in 3 dimensions, 18 individuals (9 males, 9 females)

Format

`macaques$x`: An array of dimension $7 \times 3 \times 18$

`macaques$group`: A factor indicating the sex ('m' for male and 'f' for female)

Details

In an investigation into sex differences in the crania of a species of Macaca fascicularis (a type of monkey), random samples of 9 male and 9 female skulls were obtained by Paul O'Higgins (Hull-York Medical School) (Dryden and Mardia 1993). A subset of seven anatomical landmarks was located on each cranium and the three-dimensional (3D) coordinates of each point were recorded.

Note

Dryden, I.L. and Mardia, K.V. (1998). Statistical Shape Analysis, Wiley, Chichester.

References

Dryden, I. L. and Mardia, K. V. (1993). Multivariate shape analysis. Sankhya Series A, 55, 460-480.

meteorology *meteorological data*

Description

A meteorological data include 46 records about air, soil, humidity, wind and evaporation.

Format

`meteorology$air`: A data.frame containing 3 variables: maximum, minimum and average daily air temperature

`meteorology$soil`: A data.frame containing 3 covariates: maximum, minimum and average daily soil temperature

`meteorology$humidity`: A data.frame containing 3 covariates: maximum, minimum and average daily humidity temperature,

`meteorology$wind`: a vector object record total wind, measured in miles per day

`meteorology$evaporation`: a vector object record evaporation

Details

This meteorological data containing 46 observations on five groups of variables: air temperature, soil temperature, relative humidity, wind speed as well as evaporation. Among them, maximum, minimum and average value for air temperature, soil temperature, and relative humidity are recorded. As regards to wind speed and evaporation, there are univariate numerical variables. We desire to test the independence of these five groups of variables.

`nhdist` *Distance Matrix Computation for Non-Hilbert Data*

Description

This function computes and returns the numeric distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Usage

```
nhdist(x, method = "geodesic")
```

Arguments

`x`	a numeric matrix, data frame or numeric array of dimension $k \times m \times n$ containing n samples in $k \times m$ dimension.
`method`	the distance measure to be used. This must be one of `"geodesic"`, `"compositional"`, or `"riemann"`. Any unambiguous substring can be given.

Details

Available distance measures are geodesic, compositional and riemann. Denoting any two sample in the dataset as x and y, we give the definition of distance measures as follows.

geodesic:

The shortest route between two points on the Earth's surface, namely, a segment of a great circle.

$$\arccos(x^T y), \|x\|_2 = \|y\|_2 = 1$$

compositional:

First, we apply scale transformation to it, i.e., $(x_{i1}/t, ..., x_{ip}/t_i), t_i = \sum_{d=1}^{p} x_d$. Then, apply the square root transformation to data and calculate the geodesic distance between samples.

riemann:

$k \times m \times n$ array where k = number of landmarks, m = number of dimensions and n = sample size. Detail about riemannian shape distance was given in Kendall, D. G. (1984).

Value

$n \times n$ numeric distance matrix

References

Kendall, D. G. (1984). Shape manifolds, Procrustean metrics and complex projective spaces, Bulletin of the London Mathematical Society, 16, 81-121.

Examples

```
data('bdvmf')
Dmat <- nhdist(bdvmf[['x']], method = "geodesic")

data("ArcticLake")
Dmat <- nhdist(ArcticLake[['x']], method = "compositional")

data("macaques")
Dmat <- nhdist(macaques[["x"]], method = "riemann")

# unambiguous substring also available:
Dmat <- nhdist(macaques[["x"]], method = "rie")
```

Index

ArcticLake, 2

bcor, 3, *7, 8*, *12*
bcorsis, *5*, 5
bcov, *11, 12*
bcov *(*bcor*)*, 3
bcov.test, *4, 5*, 9
bd, 13, *17*, *20, 21*
bd.gwas.test, 16
bd.test, *15*, *17*, 18
bdvmf, 22

genlung, 23

list, *7*, *11*

macaques, 24
meteorology, 24

nhdist, 25