Package ‘highDmean’

July 22, 2025

Type Package

Title Testing Two-Sample Mean in High Dimension

Version 0.1.0

Author Huaiyu Zhang, Haiyan Wang

Maintainer Huaiyu Zhang <`huaiyuzhang1988@gmail.com`>

Description Implements the high-dimensional two-sample test proposed by Zhang (2019) <`http://hdl.handle.net/2097/40235`>. It also implements the test proposed by Srivastava, Katayama, and Kano (2013) <doi:10.1016/j.jmva.2012.08.014>. These tests are particularly suitable to high dimensional data from two populations for which the classical multivariate Hotelling's T-square test fails due to sample sizes smaller than dimensionality. In this case, the ZWL and ZWLm tests proposed by Zhang (2019) <`http://hdl.handle.net/2097/40235`>, referred to as zwl_test() in this package, provide a reliable and powerful test.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Depends R (>= 3.1.0)

Imports stats

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-12 10:30:08 UTC

Contents

`buildData` *Two-sample datasets generator*

Description

This function generates simulated high dimensional two-sample data from user specified populations with given mean vectors, covariance structure, sample sizes, and dimension of each observation. It could generate the long-range dependent process proposed by Hall et al. (1998) in additional to some processes provided in arima.sim().

Usage

```
buildData(
  n,
  m,
  p,
  muX,
  muY,
  dep,
  commoncov = TRUE,
  VarScaleY = 1,
  S = 1,
  innov = function(n, ...) stats::rnorm(n, 0, 1),
  heteroscedastic = FALSE,
  het.diag
)
```

Arguments

`n`	number of observations in the 1st sample.
`m`	number of observations in the 2nd sample.
`p`	the dimensionality of the each observation. The samples from both populations should have the same dimension.
`muX`	`p` by 1 vector of component means for the 1st population.
`muY`	`p` by 1 vector of component means for the 2nd population.
`dep`	dependence structure among the `p` components for both populations. Possible options are: 'IND' for independence; 'SD' for strong dependency, AR(1) with parameter 0.9;

	'WD' for weak dependency, ARMA(2, 2) with AR parameters 0.4 and -0.1, and MA parameters 0.2 and 0.3; 'LR' for long-range dependency with parameter 0.7. For more details about the configurations, please refer to Zhang and Wang (2020).
`commoncov`	a logical indicating whether the two populations have equal covariance matrices. If FALSE, the innovations used in generating data for the 2nd population will be scaled by the square root of the value specified in VarScaleY.
`VarScaleY`	constant by which innovations are scaled in generating observations for the 2nd sample when commoncov=FALSE.
`S`	the number of data sets to simulate.
`innov`	a function used to generate the innovations, such as `innov=function(n,...) rnorm(n,0,1)`.
`heteroscedastic`	a logical indicating whether the components will be scaled by the entries in the diagonal matrix specified by `het.diag`.
`het.diag`	a `p` by `p` diagonal matrix, where the entries on the diagonal will be used to scale the component standard deviations. Only valid when `heteroscedastic` = `TRUE`.

Value

A list of `S` lists, each consisting of an `n` by `p` matrix `X`, an `m` by `p` matrix `Y`, the sample sizes, `n` and `m`, for each population, and the dimensionality `p`.

References

Hall, P., Jing, B.-Y., and Lahiri, S. N. (1998). On the sampling window method for long-range dependent data. Statistica Sinica, 8(4):1189-1204.

Examples

```
# Generate 3 two-sample datasets of dimensionality 300
# with sample sizes 45 for one sample & 60 for the other.
buildData(n = 45, m =60, p = 300,
          muX = rep(0,300), muY = rep(0,300),
          dep = 'IND', S = 3, innov = rnorm)
```

`GO_example` *An example of GO term data*

Description

A dataset containing the gene expressions for a Gene Ontology (GO) term on two phenotype groups: BCR/ABL and NEG. The id of the GO term is `GO:0000003`. The raw dataset is taken from `ALL` package. The data were preprocessed, for which the details are elaborated in Zhang and Wang (2020).

Usage

```
GO_example
```

Format

A list with two subsets of gene expression data.

X A matrix containing gene expressions for the BCR/ABL group. The row id is for patient and the column id is for gene.

Y A matrix containing gene expressions for the NEG group. The row id is for patient and the column id is for gene.

References

Zhang, H. and Wang, H. (2020). Result consistency of high dimensional two-sample tests applied to gene ontology terms with gene sets. Manuscript in review.

`highDmean` *highDmean: A package for testing of equal mean for two-sample high dimensional data*

Description

This package is an implementation of the high-dimensional two-sample test proposed by Zhang and Wang (2020) "Result consistency of high dimensional two-sample tests applied to gene ontology terms with gene sets". It also implements the SKK test proposed by Srivastava, Katayama, and Kano (2013) "A two sample test in high dimensional data." These tests are particularly suitable for high dimensional data from two populations for which the classical multivariate Hotelling's T-square test fails due to sample sizes smaller than dimensionality. In this case, the ZWL and ZWLm tests proposed by Zhang and Wang (2020), referred to as zwl_test() in this package, provide a reliable and powerful test.

highDmean functions

The function `zwl_test()` conducts the ZWL and ZWLm test of equal mean for two-sample high dimensional data provided in matrices of dimension `n` by `p` and `m` by `p`, which are random samples from two populations. It returns the value of test statistic and p-value under the null hypothesis of equal means. The `SKK_test()` performs the SKK test and returns the value of test statistic and p-value. The `buildData()` function generates simulated high-dimensional data in the two-population setting with specified sample sizes, numbers of components, covariance structure, etc., and the functions `zwl_sim()` and `SKK_sim()` return test statistic values and p-values for lists of simulated data sets generated by `buildData()`.

`rgammashift` *Random sample from shifted gamma distribution*

Description

This function generates random samples from shifted gamma distribution. That is, random samples are first generated from gamma distribution with shape parameter `shape` and scale parameter `scale` and then the mean of the gamma distribution, `shape*scale`, is subtracted from the sample.

Usage

```
rgammashift(n, shape, scale)
```

Arguments

`n`	number of observations.
`shape`	the shape parameter of gamma distribution
`scale`	the scale parameter of gamma distribution #'

Value

A vector of `n` values. It is equivalent to rgamma(n, shape, scale)- shape * scale.

Examples

```
# Generate a sample of shifted gamma observations with shape parameter 4 and scale parameter 2.
set.seed(10)
rgammashift(n = 5, shape =4, scale = 2)
# It is equivalent to
set.seed(10)
rgamma(n = 5, shape=4, scale=2)- 4 * 2
```

`SKK_sim` *Apply the SKK test to multiple simulated two-sample datasets*

Description

This function performs the SKK test of Srivastava, Katayama, and Kano(2013) on multiple high-dimensional two-sample datasets. It is useful for Monte Carlo experiments.

Usage

```
SKK_sim(DATA)
```

Arguments

`DATA`	The list of dataset lists generated by `buildData`.

Value

a dataframe, each row of which reports the values of the SKK test statistics and the p-values.

References

Srivastava, M. S., Katayama, S., and Kano, Y. (2013). A two sample test in high dimensional data. Journal of Multivariate Analysis, 114:349-358.

Examples

```
# Generate 3 simulated datasets and apply the SKK test
data <- buildData(n = 45, m =60, p = 300,
                  muX = rep(0,300), muY = rep(0,300),
                  dep = 'IND', S = 3, innov = rnorm)
SKK_sim(data)
```

`SKK_test` *High-dimensional two-sample test (SKK) proposed by Srivastava, Katayama, and Kano(2013)*

Description

This function implements the two-sample high-dimensional test proposed by Srivastava, Katayama, and Kano(2013).

Usage

```
SKK_test(X, Y)
```

Arguments

`X`	The data matrix (n by p) from the first population.
`Y`	The data matrix (m by p) from the second population.

Value

A list consisting of the values of the test statistic and p-value.

References

Srivastava, M. S., Katayama, S., and Kano, Y. (2013). A two sample test in high dimensional data. Journal of Multivariate Analysis, 114:349-358.

Examples

```
# Generate a simulated dataset and apply the SKK test
data <- buildData(n = 45, m =60, p = 300,
                 muX = rep(0,300), muY = rep(0,300),
                 dep = 'IND', S = 1, innov = rnorm)
SKK_test(data[[1]]$X, data[[1]]$Y)

# Apply the SKK test to the data for a GO term stored in GO_example
SKK_test(GO_example$X, GO_example$Y)
```

`zwl_sim` *Apply the test by Zhang and Wang (2020) to multiple simulated two-sample datasets*

Description

Apply the two-sample high-dimensional test by Zhang and Wang (2020) to multiple simulated two-sample high dimensional datasets. This function is useful for Monte Carlo experiments.

Usage

```
zwl_sim(DATA, order = 0)
```

Arguments

`DATA`	The list of dataset lists generated by `buildData`.
`order`	The order of the center correction. Possible choices are 0, 2. To use the ZWLm test, set `order = 0`; to use the ZWL test, set `order = 2`. THE ZWLm test is recommended for moderate sample sizes.

Value

A dataframe with each row consisting the values of the test statistics, p-values, Tn, and the estimate of Var(Tn).

References

Zhang, H. and Wang, H. (2020). Result consistency of high dimensional two-sample tests applied to gene ontology terms with gene sets. Manuscript in review.

Examples

```
# Generate 3 simulated two-sample datasets and apply the ZWL test
data <- buildData(n = 45, m =60, p = 300,
          muX = rep(0,300), muY = rep(0,300),
          dep = 'IND', S = 3, innov = rnorm)
zwl_sim(data, order = 2)
```

`zwl_test` *High-dimensional two-sample test proposed by Zhang and Wang (2020)*

Description

This function implements the test of equal mean for two-sample high-dimension data using the ZWL and ZWLm tests proposed by Zhang and Wang (2020).

Usage

```
zwl_test(X, Y, order = 0)
```

Arguments

`X`	The data matrix (n by p) from the first population.
`Y`	The data matrix (m by p) from the second population.
`order`	The order of center correction. Possible choices are 0, 2. To use the ZWLm test, set `order = 0`; to use the ZWL test, set `order = 2`. For moderate sample sizes, ZWLm is recommended.

Value

statistic The value of the test statistic.

pvalue The p-value of the test statistic based on the asymptotic normality established by Zhang and Wang (2020)

Tn The average of the squared univariate t-statistics.

var The estimated variance of Tn

References

Zhang, H. and Wang, H. (2020). Result consistency of high dimensional two-sample tests applied to gene ontology terms with gene sets. Manuscript in review.

Examples

```
# Generate a simulated two-sample dataset and apply the ZWL test
data <- buildData(n = 45, m =60, p = 300,
          muX = rep(0,300), muY = rep(0,300),
          dep = 'IND', S = 1, innov = rnorm)
zwl_test(data[[1]]$X, data[[1]]$Y, order = 2)

# Apply the ZWLm test to a GO term to see if the two groups are differentiately expressed.
# The data for the GO term were stored in GO_example.
zwl_test(GO_example$X, GO_example$Y, order = 0)
# Apply the ZWL test to the GO term
zwl_test(GO_example$X, GO_example$Y, order = 2)
```

Index