Package 'kerDAA'

August 22, 2023

Type Package

Title New Kernel-Based Test for Differential Association Analysis

Version 0.1.1

Author Hoseung Song [aut, cre],
Michael C. Wu [aut]

Imports mvtnorm

Maintainer Hoseung Song <hsong3@fredhutch.org>

Description A new practical method to evaluate whether relationships between two sets of high-dimensional variables are different or not across two conditions.
Song, H. and Wu, M.C. (2023)
<arXiv:2307.15268>.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-22 21:00:06 UTC

R topics documented:

`kerDAA` *New kernel-based test for differential association analysis*

Description

This package can be used to determine whether two high-dimensional samples have similar dependence relationships across two conditions.

Author(s)

Hoseung Song and Michael C. Wu

Maintainer: Hoseung Song (hsong3@fredhutch.org)

References

Song, H. and Wu, M.C. (2023). Multivariate differential association analysis. arXiv:2307.15268

See Also

kerdaa

Examples

```
# Dimension of variables.
d = 100

# The first covariance matrix
SIG = matrix(0, d, d)
for (i in 1:d) {
  for (j in 1:d) {
    SIG[i,j] = 0.4^(abs(i-j))
  }
}

# The second covariance matrix
SIG1 = matrix(0, d, d)
for (i in 1:d) {
  for (j in 1:d) {
    SIG1[i,j] = (0.4+0.5)^(abs(i-j))
  }
}

set.seed(500)

# We use 'rmvnorm' in 'mvtnorm' package to generate multivariate normally distributed samples
require(mvtnorm)
Z = rmvnorm(100, mean = rep(0,100), sigma = SIG)
X1 = Z[,1:50]
Y1 = Z[,51:100]

Z = rmvnorm(100, mean = rep(0,100), sigma = SIG1)
X2 = Z[,1:50]
Y2 = Z[,51:100]

a = kerdaa(X1, Y1, X2, Y2, perm=1000)
# output results based on the permutation and the asymptotic results
# the test statistic values can be found in a$stat_g and a$stat_l
# p-values can be found in a$pval and a$pval_perm
```

kerdaa *New kernel-based test for differential association analysis*

Description

This function provides the kernel-based differential association test.

Usage

```
kerdaa(X1, Y1, X2, Y2, perm=0)
```

Arguments

X1	The first multivariate data in the first condition.
Y1	The second multivariate data in the first condition.
X2	The first multivariate data in the second condition.
Y2	The second multivariate data in the second condition.
perm	The number of permutations performed to calculate the p-value of the test. The default value is 0, which means the permutation is not performed and only approximated p-value based on the asymptotic theory is provided. Doing permutation could be time consuming, so be cautious if you want to set this value to be larger than 10,000.

Value

Returns a list with test statistic values and p-values of the test. See below for more details.

stat_g	The value of the test statistic using the Gaussian kernel.
stat_l	The value of the test statistic using the linear kernel.
pval	The omnibus p-value using the approximated p-values of the test statistic based on asymptotic theory.
pval_perm	The omnibus p-value using the permutation p-values of the test statistic when argument 'perm' is positive.

See Also

kerDAA-package

Examples

```
# Dimension of variables.
d = 100

# The first covariance matrix
SIG = matrix(0, d, d)
```

```
for (i in 1:d) {
  for (j in 1:d) {
    SIG[i,j] = 0.4^(abs(i-j))
  }
}

# The second covariance matrix
SIG1 = matrix(0, d, d)
for (i in 1:d) {
  for (j in 1:d) {
    SIG1[i,j] = (0.4+0.5)^(abs(i-j))
  }
}

set.seed(500)
# We use 'rmvnorm' in 'mvtnorm' package to generate multivariate normally distributed samples
require(mvtnorm)
Z = rmvnorm(100, mean = rep(0,100), sigma = SIG)
X1 = Z[,1:50]
Y1 = Z[,51:100]

Z = rmvnorm(100, mean = rep(0,100), sigma = SIG1)
X2 = Z[,1:50]
Y2 = Z[,51:100]

a = kerdaa(X1, Y1, X2, Y2, perm=1000)
# output results based on the permutation and the asymptotic results
# the test statistic values can be found in a$stat_g and a$stat_l
# p-values can be found in a$pval and a$pval_perm
```

Index