Package 'SKIT'

December 24, 2020

Title SKIT

Version 0.0.2

Date 2019-11-19

Author DongHyuk Lee and Bin Zhu

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Description Conduct the Semiparametric Kernel Independence Test between two vectors when there are excess zeros.

Imports stats, utils

License GPL-2

NeedsCompilation yes

Archs x64

R topics documented:

`SKIT-package` *Semiparametric Kernel Independence Test (SKIT) with excess zeros*

Description

Conduct the test of independence between two vectors when there are excess zeros.

Details

The function `SKIT()` compute the test statistics and compute their p-values based on bootstrap samples.

Author(s)

Bin Zhu <bin.zhu@nih.gov> and DongHyuk Lee <donghyuk.lee@nih.gov>

skit — *Semiparametric Kernel Independence Test (SKIT) with excess zeros*

Description

Performs the test of independence between two vectors when there are excess zeros.

Usage

```
skit(x, y, bandwidth=NULL, nboot=1000, print=0)
```

Arguments

`x` Numeric vector possibly containing zeros.

`y` Numeric vector possibly containing zeros with same length as `x`.

`bandwidth` Bandwith parameter (see details). The default is NULL so that a default value will be computed from the data.

`nboot` Number of bootstrap samples to estimate p-values. If p-values are not desired, then set to 0. The default is 1000.

`print` 0 or 1 to print information. The default is 0.

Details

Non-numeric values in `x` or `y` will be removed from both vectors.

If `bandwidth = NULL`, then the `bandwidth` parameter will be computed for the observed data, and re-computed for each bootstrap sample. It is computed as `sigma*n^(-0.2)`, where `n = length(x)`, `sigma = max(c(sd(x[x != 0]),sd(y[y != 0])))`.

Value

A list containing the observed test statistics, bootstrap-estimated p-values, and the bandwith parameter for the observed data.

The overall test statistic $\widehat{T}$ is computed as $\widehat{T}_1 + \widehat{T}_2 + \widehat{T}_3 + \widehat{T}_4$ and for each component, p-value is estimated via bootstrap.

Examples

```
set.seed(123)

n   <- 500
b00 <- 0.09; b10 <- 0.21; b01 <- 0.21; b11 <- 0.49
id  <- sample(1:4, n, replace = TRUE, prob = c(b00, b10, b01, b11))
subn <- table(factor(id, levels = 1:4))
n2 <- subn[names(subn) == "2"]
n3 <- subn[names(subn) == "3"]
n4 <- subn[names(subn) == "4"]
Biv <- matrix(0, nrow = n, ncol = 2)
if(n2 != 0) Biv[id == 2,1] <- rnorm(n2, 0.4, 0.1)
if(n3 != 0) Biv[id == 3,2] <- rnorm(n3, 0.4, 0.1)
if(n4 != 0) Biv[id == 4,] <- cbind(rnorm(n4, 0.4, 0.1), rnorm(n4, 0.4, 0.1))
x <- Biv[,1]
```

```
y <- Biv[,2]

skit(x, y, nboot=0)
```

Index