Package ‘dHSIC’

January 4, 2019

Type Package

Title Independence Testing via Hilbert Schmidt Independence Criterion

Version 2.1

Date 2019-01-04

Author Niklas Pfister and Jonas Peters

Maintainer Niklas Pfister <pfister@stat.math.ethz.ch>

Description Contains an implementation of the
d-variable Hilbert Schmidt independence criterion
and several hypothesis tests based on it, as described
in Pfister et al. (2017) <doi:10.1111/rssb.12235>.

License GPL-3

Imports Rcpp (>= 0.12.18)

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-04 13:50:19 UTC

R topics documented:

`dHSIC-package` *Independence Testing via Hilbert Schmidt Independence Criterion*

Description

Contains an implementation of the d-variable Hilbert Schmidt independence criterion and several hypothesis tests based on it, as described in Pfister et al. (2017) <doi:10.1111/rssb.12235>.

Details

The DESCRIPTION file:

Package:	dHSIC
Type:	Package
Title:	Independence Testing via Hilbert Schmidt Independence Criterion
Version:	2.1
Date:	2019-01-04
Author:	Niklas Pfister and Jonas Peters
Maintainer:	Niklas Pfister <pfister@stat.math.ethz.ch>
Description:	Contains an implementation of the d-variable Hilbert Schmidt independence criterion and several hypothesis te
License:	GPL-3
Imports:	Rcpp (>= 0.12.18)
LinkingTo:	Rcpp

Index of help topics:

```
dHSIC-package           Independence Testing via Hilbert Schmidt
                        Independence Criterion
dhsic                   d-variable Hilbert Schmidt independence
                        criterion - dHSIC
dhsic.test              Independence test based on dHSIC
```

Author(s)

Niklas Pfister and Jonas Peters

Maintainer: Niklas Pfister <pfister@stat.math.ethz.ch>

References

Gretton, A., K. Fukumizu, C. H. Teo, L. Song, B. Sch\"olkopf and A. J. Smola (2007). A kernel statistical test of independence. In Advances in Neural Information Processing Systems (pp. 585-592).

Pfister, N., P. B\"uhlmann, B. Sch\"olkopf and J. Peters (2017). Kernel-based Tests for Joint Independence. To appear in the Journal of the Royal Statistical Society, Series B.

`dhsic` *d-variable Hilbert Schmidt independence criterion - dHSIC*

Description

The d-variable Hilbert Schmidt independence criterion (dHSIC) is a non-parametric measure of dependence between an arbitrary number of variables. In the large sample limit the value of dHSIC is 0 if the variables are jointly independent and positive if there is a dependence. It is therefore able to detect any type of dependence given a sufficient amount of data.

Usage

```
dhsic(X, Y, K, kernel = "gaussian", bandwidth = 1, matrix.input = FALSE)
```

Arguments

`X` either a list of at least two numeric matrices or a single numeric matrix. The rows of a matrix correspond to the observations of a variable. It is always required that there are an equal number of observations for all variables (i.e. all matrices have to have the same number of rows). If `X` is a single numeric matrix than one has to specify the second variable as `Y` or set `matrix.input` to "TRUE". See below for more details.

`Y` a numeric matrix if `X` is also a numeric matrix and omitted if `X` is a list.

`K` a list of the gram matrices corresponding to each variable. If `K` specified the other inputs will have no effect on the computations.

`kernel` a vector of character strings specifying the kernels for each variable. There exist two pre-defined kernels: "gaussian" (Gaussian kernel with median heuristic as bandwidth) and "discrete" (discrete kernel). User defined kernels can also be used by passing the function name as a string, which will then be matched using `match.fun`. If the length of `kernel` is smaller than the number of variables the kernel specified in `kernel[1]` will be used for all variables.

`bandwidth` a numeric value specifying the size of the bandwidth used for the Gaussian kernel. Only used if kernel="gaussian.fixed".

`matrix.input` a boolean. If `matrix.input` is "TRUE" the input `X` is assumed to be a matrix in which the columns correspond to the variables.

Details

The d-variable Hilbert Schmidt independence criterion is a direct extension of the standard Hilbert Schmidt independence criterion (HSIC) from two variables to an arbitrary number of variables. It is 0 if and only if all the variables are jointly independent. This function computes an estimator of dHSIC, which converges to the actual dHSIC in the large sample limit. It is therefore possible to detect any type of dependence in the large sample limit.

If `X` is a list with d matrices, the function computes dHSIC for the corresponding d random vectors. If `X` is a matrix and `matrix.input` is "TRUE" the functions dHSIC for the columns of `X`. If `X` is a matrix and matrix.input is "FALSE" then Y needs to be a matrix, too; in this case, the function computes the dHSIC (HSIC) for the corresponding two random vectors.

For more details see the references.

Value

A list containing the following components:

dHSIC	the value of the empirical estimator of dHSIC
time	numeric vector containing computation times. time[1] is time to compute Gram matrix and time[2] is time to compute dHSIC.
bandwidth	bandwidth used during computations. Only relevant if Gaussian kernel was used.

Author(s)

Niklas Pfister and Jonas Peters

References

Gretton, A., K. Fukumizu, C. H. Teo, L. Song, B. Sch\"olkopf and A. J. Smola (2007). A kernel statistical test of independence. In Advances in Neural Information Processing Systems (pp. 585-592).

Pfister, N., P. B\"uhlmann, B. Sch\"olkopf and J. Peters (2017). Kernel-based Tests for Joint Independence. To appear in the Journal of the Royal Statistical Society, Series B.

See Also

In order to perform hypothesis tests based on dHSIC use the function dhsic.test.

Examples

```
### Three different input methods
set.seed(0)
x <- matrix(rnorm(200),ncol=2)
y <- matrix(rbinom(100,30,0.1),ncol=1)
# compute dHSIC of x and y (x is taken as a single variable)
dhsic(list(x,y),kernel=c("gaussian","discrete"))$dHSIC
dhsic(x,y,kernel=c("gaussian","discrete"))$dHSIC
# compute dHSIC of x[,1], x[,2] and y
dhsic(cbind(x,y),kernel=c("gaussian","discrete"), matrix.input=TRUE)$dHSIC

### Using a user-defined kernel (here: sigmoid kernel)
set.seed(0)
x <- matrix(rnorm(500),ncol=1)
y <- x^2+0.02*matrix(rnorm(500),ncol=1)
sigmoid <- function(x_1,x_2){
  return(tanh(sum(x_1*x_2)))
}
dhsic(x,y,kernel="sigmoid")$dHSIC
```

`dhsic.test` *Independence test based on dHSIC*

Description

Hypothesis test for finding statistically significant evidence of dependence between several variables. Uses the d-variable Hilbert Schmidt independence criterion (dHSIC) as measure of dependence. Several types of hypothesis tests are included. The null hypothesis (H_0) is that all variables are jointly independent.

Usage

```
dhsic.test(X, Y, K, alpha = 0.05, method = "permutation",
            kernel = "gaussian", B = 1000, pairwise = FALSE,
            bandwidth = 1, matrix.input = FALSE)
```

Arguments

`X`	either a list of at least two numeric matrices or a single numeric matrix. The rows of a matrix correspond to the observations of a variable. It is always required that there are an equal number of observations for all variables (i.e. all matrices have to have the same number of rows). If `X` is a single numeric matrix than one has to specify the second variable as `Y` or set `matrix.input` to "TRUE". See below for more details.
`Y`	a numeric matrix if `X` is also a numeric matrix and omitted if `X` is a list.
`K`	a list of the gram matrices corresponding to each variable. If `K` the following inputs `X`, `Y`, `kernel`, `pairwise`, `bandwidth` and `matrix.input` will be ignored.
`alpha`	a numeric value in (0,1) specifying the confidence level of the hypothesis test.
`method`	a character string specifying the type of hypothesis test used. The available options are: "gamma" (gamma approximation based test), "permutation" (permutation test (slow)), "bootstrap" (bootstrap test (slow)) and "eigenvalue" (eigenvalue based test).
`kernel`	a vector of character strings specifying the kernels for each variable. There exist two pre-defined kernels: "gaussian" (Gaussian kernel with median heuristic as bandwidth) and "discrete" (discrete kernel). User defined kernels can also be used by passing the function name as a string, which will then be matched using `match.fun`. If the length of `kernel` is smaller than the number of variables the kernel specified in `kernel[1]` will be used for all variables.
`B`	an integer value specifying the number of Monte-Carlo iterations made in the permutation and bootstrap test. Only relevant if `method` is set to "permutation" or to "bootstrap".
`pairwise`	a logical value indicating whether one should use HSIC with pairwise comparisons instead of dHSIC. Can only be true if there are more than two variables.
`bandwidth`	a numeric value specifying the size of the bandwidth used for the Gaussian kernel. Only used if kernel="gaussian.fixed".

`matrix.input` a boolean. If `matrix.input` is "TRUE" the input `X` is assumed to be a matrix in which the columns correspond to the variables.

Details

The d-variable Hilbert Schmidt independence criterion is a direct extension of the standard Hilbert Schmidt independence criterion (HSIC) from two variables to an arbitrary number of variables. It is 0 if and only if the variables are jointly independent.

4 different statistical hypothesis tests are implemented all with null hypothesis (H_0: `X[[1]]`,...,`X[[d]]` are jointly independent) and alternative hypothesis (H_A: `X[[1]]`,...,`X[[d]]` are not jointly independent): 1. Permutation test for dHSIC: exact level, slow 2. Bootstrap test for dHSIC: pointwise asymptotic level and pointwise consistent, slow 3. Gamma approximation based test for dHSIC: only approximate, fast 4. Eigenvalue based test for dHSIC: pointwise asymptotic level and pointwise consistent, medium

The null hypothesis is rejected if `statistic` is strictly greater than `crit.value`.

If `X` is a list with d matrices, the function tests for joint independence of the corresponding d random vectors. If `X` is a matrix and `matrix.input` is "TRUE" the functions tests the independence between the columns of `X`. If `X` is a matrix and `matrix.input` is "FALSE" then `Y` needs to be a matrix, too; in this case, the function tests the (pairwise) independence between the corresponding two random vectors.

For more details see the references.

Value

A list containing the following components:

`statistic` the value of the test statistic

`crit.value` critical value of the hypothesis test. The null hypothesis (H_0: joint independence) is rejected if `statistic` is greater than `crit.value`.

`p.value` p-value of the hypothesis test, i.e. the probability that a random version of the test statistic is greater than `statistic` under the calculated null hypothesis (H_0: joint independence) based on the data.

`time` numeric vector containing computation times. `time[1]` is time to compute Gram matrix, `time[2]` is time to compute dHSIC and `time[3]` is the time to compute `crit.value` and `p.value`.

`bandwidth` bandwidth used during the computation. Only relevant if Gaussian kernel was used.

Author(s)

Niklas Pfister and Jonas Peters

References

Gretton, A., K. Fukumizu, C. H. Teo, L. Song, B. Sch\"olkopf and A. J. Smola (2007). A kernel statistical test of independence. In Advances in Neural Information Processing Systems (pp. 585-592).

Pfister, N., P. B\"uhlmann, B. Sch\"olkopf and J. Peters (2017). Kernel-based Tests for Joint Independence. To appear in the Journal of the Royal Statistical Society, Series B.

See Also

In order to only compute the test statistic without p-values, use the function `dhsic`.

Examples

```
### pairwise independent but not jointly independent (pairwise HSIC vs dHSIC)
set.seed(0)
x <- matrix(rbinom(100,1,0.5),ncol=1)
y <- matrix(rbinom(100,1,0.5),ncol=1)
z <- matrix(as.numeric((x+y)==1)+rnorm(100),ncol=1)
X <- list(x,y,z)

dhsic.test(X, method="permutation",
           kernel=c("discrete", "discrete", "gaussian"),
           pairwise=TRUE, B=1000)$p.value
dhsic.test(X, method="permutation",
           kernel=c("discrete", "discrete", "gaussian"),
           pairwise=FALSE, B=1000)$p.value
```

Index