Package 'USP'

January 20, 2025

Title U-Statistic Permutation Tests of Independence for all Data Types

Version 0.1.2

Author Thomas B. Berrett <tom.berrett@warwick.ac.uk> [aut,cre], Ioannis Kontoyiannis <yiannis@maths.cam.ac.uk> [aut], Richard J. Samworth <r.samworth@statslab.cam.ac.uk> [aut]

Maintainer Thomas B. Berrett <tom.berrett@warwick.ac.uk>

Description Implements various independence tests for discrete, continuous, and infinite-dimensional data. The tests are based on a U-statistic permutation test, the USP of Berrett, Kontoyiannis and Samworth (2020) <arXiv:2001.05513>, and shown to be minimax rate optimal in a wide range of settings. As the permutation principle is used, all tests have exact, non-asymptotic Type I error control at the nominal level.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports stats, Rdpack

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-27 09:30:21 UTC

Contents

`coeffs` *Calculate coefficients of a function's series expansion*

Description

This function is used in InfKern to produce the kernel matrix from functional data defined on the interval $[0, 1]$. For further details see Section 7.4 of (Berrett et al. 2021).

Usage

```
coeffs(X, Ntrunc)
```

Arguments

`X`	The discretised functions whose coefficients are required. This should be a matrix with one row per function, and with $Ndisc$ columns, where $Ndisc$ is the grid size of the discretisation.
`Ntrunc`	The number of coefficients that are required. The function returns coefficients 1,...,$Ntrunc$.

Value

The coefficients of X in its expansion in terms of sine functions. See (Berrett et al. 2021) for more detail.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
t=seq(from=0,to=1,length.out=1000); X=t^2
U=coeffs(X,100)[1,]; L=5
plot(t,X,type="l")
approx=rep(0,1000)
for(l in 1:L){
approx=approx+qnorm(U[l])*sqrt(2)*sin((l-1/2)*pi*t)/((l-1/2)*pi)
lines(t,approx,col=l+1)
}
```

`DiscStat` *Test statistic for dependence in contingency table*

Description

This function computes the value of the test statistic T_n measuring the strength of dependence in a contingency table. See Section 3.1 of (Berrett et al. 2021) for a definition.

Usage

```
DiscStat(freq)
```

Arguments

`freq`	Two-way contingency table whose strength of dependence is to be measured.

Value

A list containing the value of the test statistic T_n, the table of expected null counts, and the table of contributions to T_n.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
freq=r2dtable(1,rep(10,5),rep(10,5))[[1]]; DiscStat(freq)

freq=diag(1:5); DiscStat(freq)

freq=r2dtable(1,rep(10,5),rep(10,5))[[1]] + 4*diag(rep(1,5))
DiscStat(freq)
```

`FourierBasis` *Fourier basis functions*

Description

Computes the values of the one-dimensional Fourier basis functions at a vector of locations x and with a vector of frequencies m. The scaling factor of 2π is included, so that the function returns, e.g., $\sqrt{2}\cos(2\pi mx)$.

Usage

```
FourierBasis(a, m, x)
```

Arguments

`a`	Sine or cosine; $a = 0$ gives cosine and $a = 1$ gives sine.
`m`	Vector of frequencies m.
`x`	Vector of locations x.

Value

Returns the values of $\sqrt{2}\cos(2\pi mx)$.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
e=FourierBasis(1,1:100,0.01); plot(0.01*(1:100),e,type="l")
e=FourierBasis(0,1,0.01*(1:100)); plot(0.01*(1:100),e,type="l")
FourierBasis(1,1:3,0.1*(1:10))
```

`FourierKernel` *Kernel matrix for Fourier basis*

Description

Calculates the kernel matrix, described in (Berrett et al. 2021) for univariate continuous data when using the Fourier basis. This function is used in USPFourier.

Usage

```
FourierKernel(x, M)
```

Arguments

`x`	A vector in $[0, 1]^n$ for some n, containing the observations.
`M`	The maximum frequency of Fourier basis functions to compute.

Value

The kernel matrix K, to be used in independence testing.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
n=10; x=runif(n)
FourierKernel(x,5)
```

`InfKern` *Kernel for infinite-dimensional example*

Description

Function to produce the kernel matrices in the infinite dimensional example described in Section 7.4 of (Berrett et al. 2021). Here, a random function is converted to a sequence of coefficients and we use the Fourier basis on these coefficients. This function is an essential part of USPFunctional.

Usage

```
InfKern(X, Ntrunc, M)
```

Arguments

`X`	Matrix giving one of the samples to be tested. Each row corresponds to a discretised function, with each column giving the values of the functions at the corresponding grid point.
`Ntrunc`	The total number of coefficients to look at in the basis expansion of the functional data.
`M`	The maximum frequency to look at in the Fourier basis.

Value

The kernel matrix for the sample X.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
n=10  #number of observations
Ndisc=1000; t=1/Ndisc #functions represented at grid points 1/Ndisc, 2/Ndisc,...,1
X=matrix(rep(0,Ndisc*n),nrow=n)
for(i in 1:n){
 x=rnorm(Ndisc,mean=0,sd=1)
 X[i,]=cumsum(x*sqrt(t))
}
InfKern(X,2,2)
```

KernStat — *Test statistic calculated from two kernel matrices*

Description

Calculate the U-statistic measure of dependence given two kernel matrices J and K, as described in Section 7.1 of (Berrett et al. 2021). For the featured examples considered these matrices can be calculated using FourierKernel or InfKern. Alternatively, if a different basis is to be used, then the kernels can be entered separately.

Usage

```
KernStat(J, K)
```

Arguments

J	$n \times n$ kernel matrix corresponding to first sample.
K	$n \times n$ kernel matrix corresponding to second sample.

Value

Test statistic measure the strength of dependence between the two samples.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). “Optimal rates for independence testing via U-statistic permutation tests.” *Annals of Statistics, to appear*.

Examples

```
x=runif(100); y=runif(100); M=3
J=FourierKernel(x,M); K=FourierKernel(y,M)
KernStat(J,K)
```

sumbasis — *Kernel entries in infinite dimensional case*

Description

Function to calculate each entry of the kernel matrix in the infinite dimensional example described in Section 7.4 of (Berrett et al. 2021). Here, a random function is converted to a sequence of coefficients and we use the Fourier basis on these coefficients. This function is only used in the function InfKern.

Usage

```
sumbasis(Ntrunc, M, x1, x2)
```

Arguments

`Ntrunc`	The total number of coefficients to look at.
`M`	The maximum frequency to look at in the Fourier basis.
`x1`	The coefficients of the first data point.
`x2`	The coefficients of the second data point.

Value

The entry of the kernel corresponding to the two data points.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). “Optimal rates for independence testing via U-statistic permutation tests.” *Annals of Statistics, to appear*.

Examples

```
x1=runif(5); x2=runif(5); sumbasis(5,2,x1,x2)
```

`USP` *Permutation test of independence.*

Description

Carry out an independence test of the independence of two samples, give two kernel matrices J and K, as described in Section 7.1 of (Berrett et al. 2021). We calculate the test statistic and null statistics using the function KernStat, before comparing them to produce a p-value. For the featured examples considered these matrices can be calculated using FourierKernel or InfKern. Alternatively, if a different basis is to be used, then the kernels can be entered separately.

Usage

```
USP(J, K, B = 999, ties.method = "standard", nullstats = FALSE)
```

Arguments

`J`	$n \times n$ kernel matrix corresponding to first sample.
`K`	$n \times n$ kernel matrix corresponding to second sample.
`B`	The number of permutation used to calibrate the test.
`ties.method`	If "standard" then calculate the p-value as in (5) of (Berrett et al. 2021), which is slightly conservative. If "random" then break ties randomly. This preserves Type I error control.
`nullstats`	If TRUE, returns a vector of the null statistic values.

Value

Returns the p-value for this independence test and the value of the test statistic, D_n, as defined in (Berrett et al. 2021). If nullstats=TRUE is used, then the function also returns a vector of the null statistics.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
x=runif(100); y=runif(100); M=3
J=FourierKernel(x,M); K=FourierKernel(y,M)
USP(J,K,999)

n=50; r=0.6; Ndisc=1000; t=1/Ndisc
X=matrix(rep(0,Ndisc*n),nrow=n); Y=matrix(rep(0,Ndisc*n),nrow=n)
for(i in 1:n){
 x = rnorm(Ndisc, mean=0, sd= 1)
 se = sqrt(1 - r^2) #standard deviation of error
 e = rnorm(Ndisc, mean=0, sd=se)
 y = r*x + e
 X[i,] = cumsum(x*sqrt(t))
 Y[i,] = cumsum(y*sqrt(t))
}
J=InfKern(X,2,1); K=InfKern(Y,2,1)
USP(J,K,999)
```

`USP.test` *Independence test for discrete data*

Description

Carry out a permutation independence test on a two-way contingency table. The test statistic is Tn, as described in Sections 3.1 and 7.1 of (Berrett et al. 2021). This also appears as Un in (Berrett and Samworth 2021). The critical value is found by sampling null contingency tables, with the same row and column totals as the input, via Patefield's algorithm, and recomputing the test statistic.

Usage

```
USP.test(freq, B = 999, ties.method = "standard", nullstats = FALSE)
```

Arguments

`freq`	Two-way contingency table whose independence is to be tested.
`B`	The number of resampled null tables to be used to calibrate the test.

`ties.method`	If "standard" then calculate the p-value as in (5) of (Berrett et al. 2021), which is slightly conservative. If "random" then break ties randomly. This preserves Type I error control.
`nullstats`	If TRUE, returns a vector of the null statistic values.

Value

Returns the p-value for this independence test and the value of the test statistic, T_n, as defined in (Berrett et al. 2021). The third element of the list is the table of expected counts, and the final element is the table of contributions to T_n. If nullstats=TRUE is used, then the function also returns a vector of the null statistics.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). “Optimal rates for independence testing via U-statistic permutation tests.” *Annals of Statistics, to appear*.

Berrett TB, Samworth RJ (2021). “USP: an independence test that improves on Pearson’s chi-squared and the G-test.” *Submitted, available at arXiv:2101.10880*.

Examples

```
freq=r2dtable(1,rep(10,5),rep(10,5))[[1]] + 4*diag(rep(1,5))
USP.test(freq,999)

freq=diag(1:5); USP.test(freq,999)

freq=r2dtable(1,rep(10,5),rep(10,5))[[1]];
test=USP.test(freq,999,nullstats=TRUE)
plot(density(test$NullStats,from=0,
to=max(max(test$NullStats),test$TestStat)),
    xlim=c(min(test$NullStats),max(max(test$NullStats),test$TestStat)),
    main="Test Statistics")
abline(v=test$TestStat,col=2); TestStats=c(test$TestStat,test$NullStats)
abline(v=quantile(TestStats,probs=0.95),lty=2)
```

`USPFourier` *Independence test for continuous data*

Description

Performs a permutation test of independence between two univariate continuous random variables, using the Fourier basis to construct the test statistic, as described in (Berrett et al. 2021).

Usage

```
USPFourier(x, y, M, B = 999, ties.method = "standard", nullstats = FALSE)
```

Arguments

`x`	A vector containing the first sample, with each entry in $[0, 1]$.
`y`	A vector containing the second sample, with each entry in $[0, 1]$.
`M`	The maximum frequency to use in the Fourier basis.
`B`	The number of permutation to use when calibrating the test.
`ties.method`	If "standard" then calculate the p-value as in (5) of (Berrett et al. 2021), which is slightly conservative. If "random" then break ties randomly. This preserves Type I error control.
`nullstats`	If TRUE, returns a vector of the null statistic values.

Value

Returns the p-value for this independence test and the value of the test statistic, D_n, as defined in (Berrett et al. 2021). If nullstats=TRUE is used, then the function also returns a vector of the null statistics.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). "Optimal rates for independence testing via U-statistic permutation tests." *Annals of Statistics, to appear*.

Examples

```
x=runif(10); y=x^2
USPFourier(x,y,1,999)

n=100; w=2; x=integer(n); y=integer(n); m=300
unifdata=matrix(runif(2*m,min=0,max=1),ncol=2); x1=unifdata[,1]; y1=unifdata[,2]
unif=runif(m); prob=0.5*(1+sin(2*pi*w*x1)*sin(2*pi*w*y1)); accept=(unif<prob);
Data1=unifdata[accept,]; x=Data1[1:n,1]; y=Data1[1:n,2]
plot(x,y)
USPFourier(x,y,2,999)

x=runif(100); y=runif(100)
test=USPFourier(x,y,3,999,nullstats=TRUE)
plot(density(test$NullStats,from=min(test$NullStats),to=max(max(test$NullStats),test$TestStat)),
     xlim=c(min(test$NullStats),max(max(test$NullStats),test$TestStat)),main="Test Statistics")
abline(v=test$TestStat,col=2); TestStats=c(test$TestStat,test$NullStats)
abline(v=quantile(TestStats,probs=0.95),lty=2)
```

`USPFourierAdapt` *Adaptive permutation test of independence for continuous data.*

Description

We implement the adaptive version of the independence test for univariate continuous data using the Fourier basis, as described in Section 4 of (Berrett et al. 2021). This applies USPFourier with a range of values of M, and a properly corrected significance level.

Usage

```
USPFourierAdapt(x, y, alpha, B = 999, ties.method = "standard")
```

Arguments

`x`	The vector of data points from the first sample, each entry belonging to $[0, 1]$.
`y`	The vector of data points from the second sample, each entry belonging to $[0, 1]$.
`alpha`	The desired significance level of the test.
`B`	Controls the number of permutations to be used. With a sample size of n each test uses $B \log_2 n$ permutations. If $B + 1 < 1/\alpha$ then it is not possible to reject the null hypothesis.
`ties.method`	If "standard" then calculate the p-value as in (5) of (Berrett et al. 2021), which is slightly conservative. If "random" then break ties randomly. This preserves Type I error control.

Value

Returns an indicator with value 1 if the null hypothesis of independence is rejected and 0 otherwise. If the null hypothesis is rejected, the function also outputs the value of M at the which the null was rejected and the value of the test statistic.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). “Optimal rates for independence testing via U-statistic permutation tests.” *Annals of Statistics, to appear*.

Examples

```
n=100; w=2; x=integer(n); y=integer(n); m=300
unifdata=matrix(runif(2*m,min=0,max=1),ncol=2); x1=unifdata[,1]; y1=unifdata[,2]
unif=runif(m); prob=0.5*(1+sin(2*pi*w*x1)*sin(2*pi*w*y1)); accept=(unif<prob);
Data1=unifdata[accept,]; x=Data1[1:n,1]; y=Data1[1:n,2]
plot(x,y)
USPFourierAdapt(x,y,0.05,999)
```

`USPFunctional` *Independence test for functional data*

Description

We implement the permutation independence test described in (Berrett et al. 2021) for functional data taking values in $L^2([0, 1])$. The discretised functions are expressed in a series expansion, and an independence test is carried out between the coefficients of the functions, using a Fourier basis to define the test statistic.

Usage

```
USPFunctional(X, Y, Ntrunc, M, B = 999, ties.method = "standard")
```

Arguments

`X`	A matrix of the discretised functional data from the first sample. There are n rows, where n is the sample size, and Ndisc columns, where Ndisc is the grid size such that the values of each function on 1/Ndisc, 2/Ndisc, ..., 1 are given.
`Y`	A matrix of the discretised functional data from the second sample. The discretisation grid may be different to the grid used for X, if required.
`Ntrunc`	The number of coefficients to retain from the series expansions of X and Y.
`M`	The maximum frequency to use in the Fourier basis when testing the independence of the coefficients.
`B`	The number of permutations used to calibrate the test.
`ties.method`	If "standard" then calculate the p-value as in (5) of (Berrett et al. 2021), which is slightly conservative. If "random" then break ties randomly. This preserves Type I error control.

Value

A p-value for the test of the independence of X and Y.

References

Berrett TB, Kontoyiannis I, Samworth RJ (2021). “Optimal rates for independence testing via U-statistic permutation tests.” *Annals of Statistics, to appear.*

Examples

```
n=50; r=0.6; Ndisc=1000; t=1/Ndisc
X=matrix(rep(0,Ndisc*n),nrow=n); Y=matrix(rep(0,Ndisc*n),nrow=n)
for(i in 1:n){
 x = rnorm(Ndisc, mean=0, sd= 1)
 se = sqrt(1 - r^2) #standard deviation of error
 e = rnorm(Ndisc, mean=0, sd=se)
 y = r*x + e
 X[i,] <- cumsum(x*sqrt(t))
 Y[i,] <- cumsum(y*sqrt(t))
}
USPFunctional(X,Y,2,1,999)
```

Index