Package 'DRPT'

July 7, 2025

Title Density Ratio Permutation Test

Version 1.1

Description Implementation of the Density Ratio Permutation Test for testing the goodness-of-fit of a hypothesised ratio of two densities, as described in Bordino and Berrett (2025) <doi:10.48550/arXiv.2505.24529>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo Rcpp

Imports Rcpp, BiasedUrn, rootSolve, future, future.apply, Rdpack (>= 2.6)

RdMacros Rdpack

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Alberto Bordino [aut, cre] (ORCID: <`https://orcid.org/0009-0006-1556-6973`>), Thomas B. Berrett [aut] (ORCID: <`https://orcid.org/0000-0002-2005-110X`>)

Maintainer Alberto Bordino <`alberto.bordino@warwick.ac.uk`>

Repository CRAN

Date/Publication 2025-07-07 09:10:05 UTC

Contents

discrete.DRPT — *A function implementing the discrete version of the DRPT for discrete data with finite support.*

Description

A function that implements the discrete version of the DRPT for discrete data with finite support as defined in Section 2.1 in Bordino and Berrett (2025).

Usage

```
discrete.DRPT(X, Y, r, H = 99, type = "V")
```

Arguments

X	A numeric vector containing the first sample.
Y	A numeric vector containing the second sample.
r	A numeric vector of positive values specifying the hypothesised density ratio in the discrete setting.
H	An integer specifying the number of permutations to use. Defaults to 99.
type	A character string indicating the test statistic to use. See the Details section for more information. Defaults to "V".

Details

Counts for the permuted samples are drawn using `rMFNCHypergeo` from the package `BiasedUrn`. When `type="U"` the test statistic is the U-statistic (12); when `type="V"` the test statistic is the V-statistic (11); setting `type="D"` gives the test statistic (56) in Appendix B of the paper.

Value

The p-value of the DRPT as defined in (2) in Bordino and Berrett (2025).

References

Bordino A, Berrett TB (2025). “Density Ratio Permutation Tests with connections to distributional shifts and conditional two-sample testing.” arXiv:2505.24529, https://arxiv.org/abs/2505.24529.

Examples

```
n = 100; m = n
X = sample(0:3, n, prob = c(1/8, 1/8, 3/8, 3/8), replace = TRUE)
Y = sample(0:3, m, prob = c(1/43, 3/43, 16/43, 23/43), replace = TRUE)
r = c(1, 3, 3, 10)

discrete.DRPT(X,Y,r,H=19)
discrete.DRPT(X,Y,r, type = "U", H=19)
discrete.DRPT(X,Y,r, type = "D", H=19)
```

`discreteT` *Compute test statistics for the DRPT in discrete settings.*

Description

Computes the test statistics introduced in Bordino and Berrett (2025) for settings where the data support is discrete and finite.

Usage

```
discreteT(NX, NY, r, n, m, type = "V")
```

Arguments

`NX` A vector of counts for the first sample. This corresponds to the sequence $\mathrm{tot}_j - N^p_{Y,j}$ with $p = \mathrm{id}$, i.e. the identity permutation, as introduced in Section 2.1 of Bordino and Berrett (2025).

`NY` A vector of counts for the second sample. This corresponds to the sequence $N^p_{Y,j}$ with $p = \mathrm{id}$, i.e. the identity permutation, as introduced in Section 2.1 of Bordino and Berrett (2025).

`r` A numeric vector of positive values specifying the hypothesised density ratio in the discrete setting.

`n` The size of the first sample.

`m` The size of the second sample.

`type` A character string indicating which test statistic to compute. One of `"U"`, `"V"`, or `"D"`. See the Details section for more information. Defaults to `"V"`.

Details

When `type = "U"`, the U-statistic (12) is calculated. When `type = "V"`, the V-statistic (11) is computed. When `type = "D"`, the test statistic (56) from Appendix B is returned.

Value

A numeric value representing the computed test statistic.

References

Bordino A, Berrett TB (2025). “Density Ratio Permutation Tests with connections to distributional shifts and conditional two-sample testing.” arXiv:2505.24529, `https://arxiv.org/abs/2505.24529`.

Examples

```
n = 100; m = n
X = sample(0:3, n, prob = c(1/4, 1/4, 1/4, 1/4), replace = TRUE)
Y = sample(0:3, m, prob = c(1/17, 3/17, 3/17, 10/17), replace = TRUE)
r = c(1, 3, 3, 10)

NX = table(X)
NY = table(Y)

discreteT(NX, NY, r, sum(NX), sum(NY), type = "V")
discreteT(NX, NY, r, sum(NX), sum(NY), type = "D")
```

DRPT *A function implementing the Density Ratio Permutation Test based on an estimate of the shifted-MMD.*

Description

A function that implements the DRPT based on the U-statistic (12) defined in Bordino and Berrett (2025). An estimator of the shifted-MMD with kernel $k(\cdot,\cdot)$ as defined in Section 3.2 of the paper is computed using the function `shiftedMMD`, which is provided in the package.

Usage

```
DRPT(X, Y, r, kernel, H = 99, S = 50)
```

Arguments

`X` A numeric vector containing the first sample.

`Y` A numeric vector containing the second sample.

`r` A function specifying the hypothesised density ratio.

`kernel` A function defining the kernel to be used for the U-statistic.

`H` An integer specifying the number of permutations to use. Defaults to 99.

`S` An integer specifying the number of steps for the Markov-Chain defined in Algorithm 2 in Bordino and Berrett (2025). Defaults to 50.

Value

The p-value of the DRPT as defined in (2) in Bordino and Berrett (2025).

References

Bordino A, Berrett TB (2025). "Density Ratio Permutation Tests with connections to distributional shifts and conditional two-sample testing." arXiv:2505.24529, https://arxiv.org/abs/2505.24529.

Examples

```
n = 50; m = 50; d = 2
r = function(x,y) {
  return(4*x*y)
   }

gaussian.kernel = function(x, y, lambda = 1){
     return(lambda^(-d) * exp(-sum(((x - y) ^ 2) / (lambda ^ 2))))
     }

X = as.matrix(cbind(runif(n, 0, 1), runif(n, 0, 1)))
Y = as.matrix(cbind(rbeta(m, 0.5, 0.3), rbeta(m, 0.5, 0.4)))

DRPT(X,Y, r, gaussian.kernel, H=19, S=10)
DRPT(X,Y, r, gaussian.kernel, H=9)
```

shiftedMMD *A function computing an estimate of the shifted-MMD.*

Description

A function computing the U-statistic (12). This serves as an estimator of the shifted-MMD defined in Section 3.2 of Bordino and Berrett (2025).

Usage

```
shiftedMMD(X, Y, r, kernel)
```

Arguments

X	A numeric vector containing the first sample.
Y	A numeric vector containing the second sample.
r	A function specifying the hypothesised density ratio.
kernel	A function defining the kernel to be used for the U-statistic.

Value

The value of the U-statistic (12).

References

Bordino A, Berrett TB (2025). “Density Ratio Permutation Tests with connections to distributional shifts and conditional two-sample testing.” arXiv:2505.24529, https://arxiv.org/abs/2505.24529.

Examples

```
n = 250; m = 250; d = 2
r = function(x,y) {
  return(4*x*y)
   }

gaussian.kernel = function(x, y, lambda = 1){
     return(lambda^(-d) * exp(-sum(((x - y) ^ 2) / (lambda ^ 2))))
     }

X = as.matrix(cbind(runif(n, 0, 1), runif(n, 0, 1)))
Y = as.matrix(cbind(rbeta(m, 0.5, 0.3), rbeta(m, 0.5, 0.4)))

shiftedMMD(X,Y, r, gaussian.kernel)
```

`starSampler` *A function implementing the star-sampler for the DRPT.*

Description

A function implementing Algorithm 2 in Bordino and Berrett (2025).

Usage

```
starSampler(X, Y, r, H = 99, S = 50)
```

Arguments

`X`	A numeric vector containing the first sample.
`Y`	A numeric vector containing the second sample.
`r`	A function specifying the hypothesised density ratio.
`H`	An integer specifying the number of permutations to use. Defaults to 99.
`S`	An integer specifying the number of steps for the Markov-Chain defined in Algorithm 2 in Bordino and Berrett (2025). Defaults to 50.

Value

A list of $H + 1$ rearrangements of the whole sample. The first element of the list is the original dataset. The other H elements are permutations of the original dataset, where permutations are generated using Algorithm 2 in the paper.

References

Bordino A, Berrett TB (2025). “Density Ratio Permutation Tests with connections to distributional shifts and conditional two-sample testing.” arXiv:2505.24529, https://arxiv.org/abs/2505.24529.

Examples

```
n = 250; m = n
r = function(x,y) {
  return(4*x*y)
   }

X = as.matrix(cbind(runif(n, 0, 1), runif(n, 0, 1)))
Y = as.matrix(cbind(rbeta(m, 0.5, 0.3), rbeta(m, 0.5, 0.4)))

starSampler(X, Y, r, H = 3, S = 20)
```

Index