Package ‘BWStest’

October 18, 2018

Type Package
Maintainer Steven E. Pav <shabbychef@gmail.com>
Version 0.2.2
Date 2018-10-17
License LGPL-3
Title Baumgartner Weiss Schindler Test of Equal Distributions

BugReports https://github.com/shabbychef/BWStest/issues


Imports memoise, Rcpp (>= 0.12.3)
LinkingTo Rcpp
Suggests testthat

RoxygenNote 6.0.1

URL https://github.com/shabbychef/BWStest

Collate 'bws_test.r' 'BWStest.r' 'RcppExports.R' 'zzz_murakami_dist.r'

NeedsCompilation yes

Author Steven E. Pav [aut, cre] (<https://orcid.org/0000-0002-4197-6195>)
Repository CRAN
Date/Publication 2018-10-18 11:40:11 UTC

R topics documented:

- BWStest-package
- BWStest-NEWS
- bws_cdf
- bws_stat
Description

Baumgartner Weiss Schindler test.

Background

The Baumgartner Weiss Schindler test is a two sample test of the null that the samples come from the same probability distribution, similar to the Kolmogorov-Smirnov, Wilcoxon, and Cramer-Von Mises tests. It is similar to the Cramer-Von Mises test in that it estimates the square norm of the difference in CDFs of the two samples. However, the Baumgartner Weiss Schindler test weights the integral by the variance of the difference in CDFs, "emphasizing the tails of the distributions, which increases the power of the test for a lot of applications."

Legal Mumbo Jumbo

BWStest is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


**bws_cdf**

CDF of the Baumgartner-Weiss-Schindler test under the null.

**Description**

Computes the CDF of the Baumgartner-Weiss-Schindler test statistic under the null hypothesis of equal distributions.

**Usage**

```r
bws_cdf(b, max.j = 5L, lower_tail = TRUE)
```
Arguments

- **b**: a vector of BWS test statistics.
- **max_j**: the maximum value of j to take in the approximate computation of the CDF via equation (2.5). Baumgartner *et. al.* claim that a value of 3 is sufficient.
- **lower_tail**: boolean, when `TRUE` returns $\Psi$, otherwise compute the upper tail, $1 - \Psi$, which is more useful for hypothesis tests.

Details

Given value $b$, computes the CDF of the BWS statistic under the null, denoted as $\Psi(b)$ by Baumgartner *et al.* The CDF is computed from equation (2.5) via numerical quadrature.

The expression for the CDF contains the integral

$$
\int_0^1 \frac{1}{\sqrt{\pi^4(1-r)}} \exp \left(\frac{rb}{8} - \frac{\pi^2(4j + 1)^2}{8rb}\right) dr
$$

By making the change of variables $x = 2r - 1$, this can be re-expressed as an integral of the form

$$
\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) dx,
$$

for some function $f(x)$ involving $b$ and $j$. This integral can be approximated via Gaussian quadrature using Chebyshev nodes (of the first kind), which is the approach we take here.

Value

A vector of the CDF of $b$, $\Psi(b)$.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also

`bws_stat`, `bws_test`

Examples

```r
# do it 500 times
set.seed(123)
bvals <- replicate(500, bws_stat(rnorm(50), rnorm(50)))
pvals <- bws_cdf(bvals)
# these should be uniform!
## Not run:
```
bws_stat

Compute the test statistic of the Baumgartner-Weiss-Schindler test.

Description

Compute the Baumgartner-Weiss-Schindler test statistic.

Usage

bws_stat(x, y)

Arguments

x  
a vector.

y  
a vector.

Details

Given vectors $X$ and $Y$, computes $B_X$ and $B_Y$ as described by Baumgartner et al., returning their average, $B$. The test statistic approximates the variance-weighted square norm of the difference in CDFs of the two distributions. For sufficiently large sample sizes (more than 20, say), under the null the test statistic approaches the asymptotic value computed in `bws_cdf`.

The test value is an approximation of

$$
\hat{B} = \frac{mn}{m+n} \int_0^1 \frac{1}{z(1-z)} (F_X(z) - F_Y(z))^2 \, dz,
$$

where $m$ ($n$) is the number of elements in $X$ ($Y$), and $F_X(z)$ ($F_Y(z)$) is the CDF of $X$ ($Y$).

The test statistic is based only on the ranks of the input. If the same monotonic transform is applied to both vectors, the result should be unchanged. Moreover, the test is inherently two-sided, so swapping $X$ and $Y$ should also leave the test statistic unchanged.

Value

The BWS test statistic, $B$. 

plot(ecdf(pvals))

## End(Not run)

# compare to Table 1 of Baumgartner et al.
bvals <- c(1.933,2.493,3.076,3.880,4.500,5.990)
tab1v <- c(0.9,0.95,0.975,0.990,0.995,0.999)
pvals <- bws_cdf(bvals,lower_tail=TRUE)
show(data.frame(B=bvals,BWS.psi=tab1v,our.psi=pvals))
Author(s)
Steven E. Pav <shabbychef@gmail.com>

References

See Also
bws_cdf, bws_test

Examples

```r
set.seed(1234)
x <- runif(1000)
y <- runif(100)
bval <- bws_stat(x, y)
# check a monotonic transform:
ftrans <- function(x) { log(1 + x) }
bval2 <- bws_stat(ftrans(x), ftrans(y))
stopifnot(all.equal(bval, bval2))
# check commutivity
bval3 <- bws_stat(y, x)
stopifnot(all.equal(bval, bval3))
```

bws_test

Perform the Baumgartner-Weiss-Schindler hypothesis test.

Description
Perform the Baumgartner-Weiss-Schindler hypothesis test.

Usage

```r
bws_test(x, y, method = c("default", "BWS", "Neuhauser", "B1", "B2", "B3", "B4", "B5"), alternative = c("two.sided", "greater", "less"))
```

Arguments

- `x`: a vector of the first sample.
- `y`: a vector of the first sample.
- `method`: a character string specifying the test statistic to use. should be one of the following:
  - `default` This is “Hobson’s choice”, which uses the classical BWS test for two-sided alternative, but Neuhauser for one sided alternatives.
Use the classical BWS test.

**Neuhauser**
Use Neuhauser’s test.

**B1**
Use Murakami’s $B_1$ test.

**B2**
Use Murakami’s $B_2$ test, which is exactly Neuhauser’s test.

**B3**
Use Murakami’s $B_3$ test.

**B4**
Use Murakami’s $B_4$ test.

**B5**
Use Murakami’s $B_5$ test.

Only Neuhauser’s test supports one-sided alternatives.

**alternative**
a character string specifying the alternative hypothesis, must be one of “two.sided” (default), “greater” or “less”. You can specify just the initial letter. “greater” corresponds to testing whether the survival function of $x$ is greater than that of $y$; equivalently one can think of this as $x$ being ‘greater’ than $y$ in the sense of first order stochastic dominance.

**Value**
Object of class `htest`, a list of the test statistic, the p-value, and the method noted.

**Note**
The code will happily compute Murakami’s $B_3$ through $B_5$ for large sample sizes, even though nominal coverage is not achieved. A warning will be thrown. User assumes all risk relying on results from this function.

**Author(s)**
Steven E. Pav <shabbychef@gmail.com>

**References**

**See Also**
`bws_test, bws_stat, murakami_stat, murakami_cdf`

**Examples**

```r
# under the null
set.seed(123)
x <- rnorm(100)
y <- rnorm(100)
hval <- bws_test(x,y)

# under the alternative
set.seed(123)
x <- rnorm(100)
```
```r
y <- rnorm(100, mean=1.0)
hval <- bws_test(x, y)
show(hval)
stopifnot(hval$p.value < 0.05)

# under the alternative with a one sided test.
set.seed(123)
x <- rnorm(100)
y <- rnorm(100, mean=0.7)
hval <- bws_test(x, y, alternative='less')
show(hval)
stopifnot(hval$p.value < 0.01)

hval <- bws_test(x, y, alternative='greater')
stopifnot(hval$p.value > 0.99)

hval <- bws_test(x, y, alternative='two.sided')
stopifnot(hval$p.value < 0.05)
```

---

### murakami_cdf

**Murakami test statistic distribution.**

#### Description

Estimates the CDF of the Murakami test statistics via permutations.

#### Usage

```r
murakami_cdf(B, n1, n2, flavor = 0L, lower_tail = TRUE)
```

#### Arguments

- `B`: the Murakami test statistic or a vector of the same.
- `n1`: number of elements in the first sample.
- `n2`: number of elements in the second sample.
- `flavor`: the 'flavor' of the test statistic. See `murakami_stat`.
- `lower_tail`: boolean, when TRUE returns the CDF, \( \Psi \), otherwise compute the upper tail, \( 1 - \Psi \), which is potentially more useful for hypothesis tests.

#### Details

Given the Murakami test statistic \( B_j \) for \( 0 \leq j \leq 5 \), computes the CDF under the null that the two samples come from the same distribution. The CDF is computed by permutation test and memoization.

#### Value

A vector of the same size as `B` of the CDF under the null.
Note

the CDF is approximately computed by evaluating the permutations up to some reasonably small sample size (currently the cutoff is 9). When larger sample sizes are used, the distribution of the test statistic may not converge. This is apparently seen in flavors 3 through 5.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also

murakami_stat.

Examples

# basic usage:
xv <- seq(0,4,length.out=101)
yv <- murakami_cdf(xv, n1=8, n2=6, flavor=1L)
plot(xv,yv)
zv <- bws_cdf(xv)
lines(xv,zv,col='red')

# check under the null:
## Not run:
flavor <- 1L
n1 <- 8
n2 <- 8
set.seed(1234)
Bvals <- replicate(2000,murakami_stat(rnorm(n1),rnorm(n2),flavor))
# should be uniform:
plot(ecdf(murakami_cdf(Bvals,n1,n2,flavor)))

## End(Not run)

---

murakami_stat | Compute Murakami’s test statistic.

Description
Compute one of the modified Baumgartner-Weiss-Schindler test statistics proposed by Murakami, or Neuhauser.

Usage
murakami_stat(x, y, flavor = 0L)
murakami_stat_perms(nx, ny, flavor = 0L)

Arguments

- **x**: a vector of the first sample.
- **y**: a vector of the second sample.
- **flavor**: which ‘flavor’ of test statistic.
- **nx**: the length of x, the first sample.
- **ny**: the length of y, the second sample.

Details
Given vectors $X$ and $Y$, computes $B_{jX}$ and $B_{jY}$ for some $j$ as described by Murakami and by Neuhauser, returning either their their average or their average distance. The test statistics approximate the weighted square norm of the difference in CDFs of the two distributions.

The test statistic is based only on the ranks of the input. If the same monotonic transform is applied to both vectors, the result should be unchanged.

The various ‘flavor’s of test statistic are:

- **0** The statistic of Baumgartner-Weiss-Schindler.
- **1** Murakami’s $B_1$ statistic, from his 2006 paper.
- **2** Neuhauser’s difference statistic, denoted by Murakami as $B_2$ in his 2012 paper.
- **3** Murakami’s $B_3$ statistic, from his 2012 paper.
- **4** Murakami’s $B_4$ statistic, from his 2012 paper.
- **5** Murakami’s $B_5$ statistic, from his 2012 paper, with a log weighting.
**Value**

The BWS test statistic, $B_j$. For `murakami_stat_perms`, a vector of the test statistics for all permutations of the input.

**Note**

NA and NaN are not yet dealt with!

**Author(s)**

Steven E. Pav <shabbychef@gmail.com>

**References**


**See Also**

`bws_stat`

**Examples**

```r
set.seed(1234)
x <- runif(1000)
y <- runif(100)
bval <- murakami_stat(x,y,1)

## Not run:
x <- 6
y <- 5
# monte carlo
set.seed(1234)
repli <- replicate(3000,murakami_stat(rnorm(nx),rnorm(ny),0L))
```
# under the null, perform the permutation test:
allem <- murakami_stat perms(nx, ny, 0L)
plot(ecdf(allem))
lines(ecdf(repli), col='red')

## End(Not run)
Index

*Topic htest
  bws_test, 6
*Topic package
  BWStest-package, 2

bws_cdf, 3, 5, 6
bws_stat, 4, 5, 7, 11
bws_test, 4, 6, 6, 7
BWStest-NEWS, 3
BWStest-package, 2

murakami_cdf, 7, 8
murakami_stat, 7–9, 10
murakami_stat_perms (murakami_stat), 10