Package ‘cramer’

January 5, 2019

Version 0.9-3
Date 2019-01-05
Title Multivariate Nonparametric Cramer-Test for the Two-Sample-Problem
Author Carsten Franz <carsten.franz@gmail.com>
Maintainer Carsten Franz <carsten.franz@gmail.com>
Depends R (>= 0.65), boot
Description Provides R routine for the so called two-sample Cramer-Test. This nonparametric two-sample-test on equality of the underlying distributions can be applied to multivariate data as well as univariate data. It offers two possibilities to approximate the critical value both of which are included in this package.
License GPL (>= 2)
Repository CRAN
NeedsCompilation no
Date/Publication 2019-01-05 10:20:03 UTC

R topics documented:
cramer-package .............................................................. 1

Index

| cramer-package | Cramer-Test for uni- and multivariate two-sample-problem |

Description

Perform Cramér-test for two-sample-problem. Both univariate and multivariate data is possible. For calculation of the critical value Monte-Carlo-bootstrap-methods and eigenvalue-methods are available. For the bootstrap access ordinary and permutation methods can be chosen as well as the number of bootstrap-replicates taken.
Usage

cramer.test(x, y, conf.level = 0.95, replicates = 1000, 
        sim = "ordinary", just.statistic = FALSE, 
        kernel = "phiCramer", maxM = 2^14, K = 160)

Arguments

x
First set of observations. Either in vector form (univariate) or in a matrix with 
one observation per row (multivariate).

y
Second set of observations. Same dimension as x.

conf.level
Confidence level of test. The default is conf.level = 0.95.

sim
Type of Monte-Carlo-bootstrap method or eigenvalue method. Possible values 
are "ordinary" (default) for normal Monte-Carlo-bootstrap, "permutation" 
for a permutation Monte-Carlo-bootstrap or "eigenvalue" for bootstrapping 
the limit distribution, evaluating the (approximate) eigenvalues being the weights 
of the limiting chisquared-distribution and using the critical value of this approx-
imation (calculated via fast fourier transform). This method is especially good if 
the dataset is too large to perform Monte-Carlo-bootstrapping (although it must 
not be too large so the matrix eigenvalue problem can still be solved).

replicates
Number of bootstrap-replicates taken to obtain critical value. The default is 
replicates = 1000. When using the eigenvalue method, this variable is unused.

maxM
Gives the maximum number of points used for the fast fourier transform. When 
using Monte-Carlo-bootstrap methods, this variable is unused.

K
Gives the upper value up to which the integral for the calculation of the distribu-
tion function out of the characteristic function (Gurlands formula) is evaluated. 
The default is 160. Careful: When increasing K it is necessary to increase maxM 
as well since the resolution of the points where the distribution function is cal-
culated is

\[
\frac{2\pi}{K}
\]

Thus, if just K is increased the maximum value, where the distribution function 
is calculated is lower. When using Monte-Carlo-bootstrap methods, this variable 
is unused.

just.statistic
Boolean variable. If TRUE just the value of the Cramér-statistic is calculated and 
no bootstrap-replicates are produced.

kernel
Character-string giving the name of the kernel function. The default is "phiCramer" 
which is the Cramér-test included in earlier versions of this package and which 
are used in the paper of Baringhaus and the author mentioned below. It is possible 
to use user-defined kernel functions here. The functions needs to be able to deal 
with matrix arguments. Kernel functions need to be defined on the positive real 
line with value 0 at 0 and have a nonconstant completely monotone first deriva-
tive. An example is show in the Examples section below. Build-in functions are 
"phiCramer", "phiBahr", "phiLog", "phiFracA" and "phiFracB".
Details

The Cramér-statistic is given by

\[ T_{m,n} = \frac{mn}{m+n} \left( \frac{2}{mn} \sum_{i,j} \phi(\|\vec{X}_i - \vec{Y}_j\|^2) - \frac{1}{m^2} \sum_{i,j=1}^m \phi(\|\vec{X}_i - \vec{X}_j\|^2) - \frac{1}{n^2} \sum_{i,j=1}^n \phi(\|\vec{Y}_i - \vec{Y}_j\|^2) \right), \]

where \( \phi(\cdot) \) is the kernel function mentioned in the Parameters section. The proof that the Monte-Carlo-Bootstrap and eigenvalue methods work is given in the reference listed below. Other build-in kernel functions are

\( \phi_{\text{Cramer}}(z) = \sqrt{z}/2 \) (recommended for location alternatives),

\( \phi_{\text{Bahr}}(z) = 1 - \exp(-z/2) \) (recommended for dispersion as well as location alternatives),

\( \phi_{\text{log}}(z) = \log(1 + z) \) (preferably for location alternatives),

\( \phi_{\text{FracA}}(z) = 1 - \frac{1}{1 + z} \) (preferably for dispersion alternatives) and

\( \phi_{\text{FracB}}(z) = 1 - \frac{1}{(1 + z)^2} \) (also for dispersion alternatives). A further analysis of the test performance for these kernels will be included in a further publication. The idea of using this statistic is due to L. Baringhaus, University of Hanover.

Value

The returned value is an object of class "cramertest", containing the following components:

- **method**: Describing the test in words.
- **d**: Dimension of the observations.
- **m**: Number of \( x \) observations.
- **n**: Number of \( y \) observations.
- **statistic**: Value of the Cramér-statistic for the given observations.
- **conf.level**: Confidence level for the test.
- **crit.value**: Critical value calculated by bootstrap method, eigenvalue method, respectively. When using the eigenvalue method, the distribution under the hypothesis will be interpolated linearly.
| **p.value** | Estimated p-value of the test. |
| **result** | Contains 1 if the hypothesis of equal distributions should not be accepted and 0 otherwise. |
| **sim** | Method used for obtaining the critical value. |
| **replicates** | Number of bootstrap-replicates taken. |
| **ev** | Contains eigenvalues and eigenfunctions when using the eigenvalue-method to obtain the critical value. |
| **hypdist** | Contains the via fft reconstructed distribution function under the hypothesis. $x$ contains the x-values and $Fx$ the values of the distribution function at the positions. |

**References**

The test and its properties is described in:


The test of Bahr so far is only mentioned in:


The eigenvalue method will be described in a forthcoming article.

**Examples**

```r
# comparison of two univariate normal distributions
x <- rnorm(20, mean=0, sd=1)
y <- rnorm(50, mean=0.5, sd=1)
cramer.test(x, y)

# comparison of two multivariate normal distributions with permutation test:
# library "MASS" for multivariate routines (included in package "VR")

# library(MASS)
# x <- mvrnorm(n=20, mu=c(0,0), Sigma=diag(c(1,1)))
# y <- mvrnorm(n=50, mu=c(0.3,0), Sigma=diag(c(1,1)))
# cramer.test(x, y, sim="permutation")

# comparison of two univariate normal distributions with Bahrs Kernel
phiBahr <- function(x) return(1-exp(-x/2))
x <- rnorm(20, mean=0, sd=1)
y <- rnorm(50, mean=0, sd=2)
cramer.test(x, y, sim="eigenvalue", kernel="phiBahr")
```
Index

*Topic htest
  cramer-package, 1
*Topic multivariate
  cramer-package, 1
*Topic nonparametric
  cramer-package, 1

cramer (cramer-package), 1
cramer-package (cramer-package), 1
cramer-package, 1
cramer.test (cramer-package), 1

phiBahr (cramer-package), 1
phiCramer (cramer-package), 1
phiFracA (cramer-package), 1
phiFracB (cramer-package), 1
phiLog (cramer-package), 1
print.cramertest (cramer-package), 1