Package ‘Rfolding’

October 12, 2022

Type Package
Title The Folding Test of Unimodality
Version 1.0
Description The basic algorithm to perform the folding test of unimodality.
  Given a dataset \(\mathbf{X}\) (\(d\) dimensional, \(n\) samples), the test checks whether the
  distribution of the data are rather unimodal or rather multimodal. This
  package stems from the following research publication:
  Siffer Alban, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouët.
  ``Are your data gathered?'' In Proceedings of the 24th ACM SIGKDD International
Encoding UTF-8
License GPL-3
LazyData true
RoxygenNote 6.1.0
Suggests testthat,MASS,knitr,rmarkdown
Imports stats
VignetteBuilder knitr
NeedsCompilation no
Author Alban Siffer [aut, cre],
  Amossys [cph, fnd]
Maintainer Alban Siffer <alban.siffer@irisa.fr>
Repository CRAN
Date/Publication 2018-09-30 13:10:12 UTC

R topics documented:

- folding.ratio .................................................. 2
- folding.statistics ............................................. 2
- folding.test .................................................. 3
- folding.test.bound .......................................... 4
folding.ratio

**Description**

Computes the folding ratio of the input data

**Usage**

```r
folding.ratio(X)
```

**Arguments**

- `X`: nxd matrix (n observations, d dimensions)

**Value**

the folding ratio

**Examples**

```r
X = matrix(runif(n = 1000, min = 0., max = 1.), ncol = 1)
phi = folding.statistics(X)
```

---

folding.statistics

**Description**

Computes the folding statistics of the input data

**Usage**

```r
folding.statistics(X)
```

**Arguments**

- `X`: nxd matrix (n observations, d dimensions)

**Value**

the folding statistics
Examples

```r
library(MASS)
mu = c(0,0)
Sigma = matrix(c(1,0.5,0.5,0.5,1), ncol = 2)
X = mvrnorm(n = 5000, mu = mu, Sigma = Sigma)
Phi = folding.statistics(X)
```

---

**folding.test**

*Perform the folding test of unimodality*

Description

Perform the folding test of unimodality

Usage

```r
folding.test(X)
```

Arguments

- `X`: $nxd$ matrix (n observations, d dimensions)

Value

1 if unimodal, 0 if multimodal

Examples

```r
library(MASS)
n = 10000
d = 3
mu = c(0,0,0)
Sigma = matrix(c(1,0.5,0.5,0.5,1,0.5,0.5,0.5,1), ncol = d)
X = mvrnorm(n = n, mu = mu, Sigma = Sigma)
m = folding.test(X)
```
folding.test.bound  
Computes the confidence bound for the significance level \( p \)

**Description**
Computes the confidence bound for the significance level \( p \)

**Usage**
folding.test.bound(n, d, p)

**Arguments**
- \( n \)  sample size
- \( d \)  dimension
- \( p \)  significance level (between 0 and 1, the lower, the more significant)

**Value**
the confidence bound \( q \) (the bounds are \( 1-q \) and \( 1+q \))

**Examples**
n = 2000  # number of observations
d = 2   # 2 dimensional data
p = 0.05  # we want the bound at the level 0.05 (classical p-value)
q = folding.test.bound(n, d, p)

---

folding.test.pvalue  
Computes the p-value of the folding test

**Description**
Computes the p-value of the folding test

**Usage**
folding.test.pvalue(Phi, n, d)

**Arguments**
- \( \Phi \)  the folding statistics
- \( n \)  sample size
- \( d \)  dimension
**pivot.approx**

**Value**

the p-value (the lower, the more significant)

**Examples**

```r
library(MASS)
n = 5000
d = 2
mu = c(0,0)
Sigma = matrix(c(1,0.5,0.5,0.5), ncol = d)
X = mvrnorm(n = n, mu = mu, Sigma = Sigma)
Phi = folding.statistics(X)
p = folding.test.pvalue(Phi,n,d)
```

---

**pivot.approx**  
*Computes the pivot $s_2$ (approximate pivot)*

**Description**

Computes the pivot $s_2$ (approximate pivot)

**Usage**

```r
pivot.approx(X)
```

**Arguments**

- `X` nxd matrix (n observations, d dimensions)

**Value**

the approximate pivot

**Examples**

```r
library(MASS)
mu = c(0,0)
Sigma = matrix(c(1,0.5,0.5,0.5), ncol = 2)
X = mvrnorm(n = 5000, mu = mu, Sigma = Sigma)
Phi = pivot.approx(X)
```
Index

folding.ratio, 2
folding.statistics, 2
folding.test, 3
folding.test.bound, 4
folding.test.pvalue, 4

pivot.approx, 5