Package: QuadratiK (via r-universe)

February 12, 2025

Type Package

Title Collection of Methods Constructed using Kernel-Based Quadratic Distances

Version 1.1.3

Maintainer Giovanni Saraceno <giovanni.saraceno@unipd.it>

Description It includes test for multivariate normality, test for uniformity on the d-dimensional Sphere, non-parametric two- and k-sample tests, random generation of points from the Poisson kernel-based density and clustering algorithm for spherical data. For more information see Saraceno G., Markatou M., Mukhopadhyay R. and Golzy M. (2024) <doi:10.48550/arXiv.2402.02290> Markatou, M. and Saraceno, G. (2024) <doi:10.48550/arXiv.2407.16374>, Ding, Y., Markatou, M. and Saraceno, G. (2023) <doi:10.5705/ss.202022.0347>, and Golzy, M. and Markatou, M. (2020) <doi:10.1080/10618600.2020.1740713>.

License GPL (>= 3)

URL https://CRAN.R-project.org/package=QuadratiK, https://github.com/ropensci/QuadratiK/, https://docs.ropensci.org/QuadratiK/

BugReports https://github.com/ropensci/QuadratiK/issues

Depends R (>= 3.5.0)

Imports parallel, doParallel, foreach, ggplot2, ggpubr, methods, moments, mvtnorm, Rcpp, RcppEigen, rlecuyer, sn, stats, rrcov, scatterplot3d

Suggests knitr, rmarkdown, roxygen2, testthat (>= 3.0.0), rgl, sphunif, circular, cluster, clusterRepro, mclust, Tinflex, movMF

LinkingTo Rcpp, RcppEigen

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list (markdown = TRUE, roclets = c (``namespace", ``rd",
``srr::srr_stats_roclet"))

RoxygenNote 7.3.2

Language en-US

Config/pak/sysreqs cmake make libicu-dev

Repository https://ropensci.r-universe.dev

RemoteUrl https://github.com/ropensci/QuadratiK

RemoteRef main

RemoteSha cbbfdd79f9205570eb879abdb56b7205095a2769

Contents

`QuadratiK-package` *Collection of Methods Constructed using the Kernel-Based Quadratic Distances*

Description

Collection of Methods Constructed using the Kernel-Based Quadratic Distances

`QuadratiK` provides the first implementation, in R and Python, of a comprehensive set of goodness-of-fit tests and a clustering technique for d-dimensional spherical data $d \geq 2$ using kernel-based quadratic distances. It includes:

- **Goodness-of-Fit Tests**: The software implements one, two, and k-sample tests for goodness of fit, offering an efficient and mathematically sound way to assess the fit of probability distributions. Our tests are particularly useful for large, high dimensional data sets where the assessment of fit of probability models is of interest. Specifically, we offer tests for normality, as well as two- and k-sample tests, where testing equality of two or more distributions is of interest, that is $H_0 : F_1 = F_2$ and $H_0 : F_1 = \ldots = F_k$ respectively. The proposed tests perform well in terms of level and power for contiguous alternatives, heavy tailed distributions and in higher dimensions.
  Expanded capabilities include supporting tests for uniformity on the d-dimensional Sphere based on the Poisson kernel, exhibiting excellent results especially in the case of multimodal distributions.
- **Poisson kernel-based distribution (PKBD)**: the package offers functions for computing the density value and for generating random samples from a PKBD. The Poisson kernel-based densities are based on the normalized Poisson kernel and are defined on the d-dimensional unit sphere. Given a vector $\mu \in \mathcal{S}^{d-1}$, and a parameter ρ such that $0 < \rho < 1$, the probability density function of a d-variate Poisson kernel-based density is defined by:

$$f(\mathbf{x}|\rho, \mu) = \frac{1 - \rho^2}{\omega_d ||\mathbf{x} - \rho\mu||^d},$$

  where μ is a vector orienting the center of the distribution, ρ is a parameter to control the concentration of the distribution around the vector μ and it is related to the variance of the distribution. Furthermore, $\omega_d = 2\pi^{d/2}[\Gamma(d/2)]^{-1}$ is the surface area of the unit sphere in $\mathbb{R}^d$ (see Golzy and Markatou, 2020).
- **Clustering Algorithm for Spherical Data**: the package incorporates a unique clustering algorithm specifically tailored for d-dimensional spherical data and it is especially useful in the presence of noise in the data and the presence of non-negligible overlap between clusters. This algorithm leverages a mixture of Poisson kernel-based densities on the Sphere, enabling effective clustering of spherical data or data that has been spherically transformed.
- **Additional Features**: Alongside these functionalities, the software includes additional graphical functions, aiding users in validating and representing the cluster results as well as enhancing the interpretability and usability of the analysis.

For an introduction to `QuadratiK` see the vignette Introduction to the QuadratiK Package.

Details

The work has been supported by Kaleida Health Foundation and the National Science Foundation.

Note

The `QuadratiK` package is also available in Python on PyPI https://pypi.org/project/QuadratiK/ and also as a Dashboard application. Usage instruction for the Dashboard can be found at <https://quadratik.readthedocs.io/en dashboard_application_usage.html>.

Author(s)

Giovanni Saraceno, Marianthi Markatou, Raktim Mukhopadhyay, Mojgan Golzy

Maintainer: Giovanni Saraceno <`giovanni.saracen@unipd.it`>

References

Saraceno, G., Markatou, M., Mukhopadhyay, R. and Golzy, M. (2024). Goodness-of-Fit and Clustering of Spherical Data: the QuadratiK package in R and Python. arXiv preprint arXiv:2402.02290.

Ding, Y., Markatou, M. and Saraceno, G. (2023). "Poisson Kernel-Based Tests for Uniformity on the d-Dimensional Sphere." Statistica Sinica. doi: doi:10.5705/ss.202022.0347.

Golzy, M. and Markatou, M. (2020) Poisson Kernel-Based Clustering on the Sphere: Convergence Properties, Identifiability, and a Method of Sampling, Journal of Computational and Graphical Statistics, 29:4, 758-770, DOI: 10.1080/10618600.2020.1740713.

Markatou, M. and Saraceno, G. (2024). "A Unified Framework for Multivariate Two- and k-Sample Kernel-based Quadratic Distance Goodness-of-Fit Tests."
https://doi.org/10.48550/arXiv.2407.16374

See Also

Useful links:

- https://CRAN.R-project.org/package=QuadratiK
- https://github.com/ropensci/QuadratiK/
- https://docs.ropensci.org/QuadratiK/
- Report bugs at https://github.com/ropensci/QuadratiK/issues

---

`breast_cancer` *Breast Cancer Wisconsin (Diagnostic)*

Description

The `breast_cancer` Wisconsin data has 569 rows and 31 columns. The first 30 variables report the features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The last column indicates the class labels (Benign = 0 or Malignant = 1).

Usage

```
breast_cancer
```

Format

A data frame of 569 observations and 31 variables.

Source

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

References

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993, July). Nuclear feature extraction for breast tumor diagnosis. In Biomedical image processing and biomedical visualization (Vol. 1905, pp. 861-870). SPIE.

Examples

```
data(breast_cancer)
summary(breast_cancer)
```

`dpkb` *The Poisson kernel-based Distribution (PKBD)*

Description

The Poisson kernel-based densities are based on the normalized Poisson kernel and are defined on the $(d-1)$-dimensional unit sphere. Given a vector $\mu \in \mathcal{S}^{d-1}$, where $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : ||x|| = 1\}$, and a parameter ρ such that $0 < \rho < 1$, the probability density function of a d-variate Poisson kernel-based density is defined by:

$$f(\mathbf{x}|\rho, \mu) = \frac{1-\rho^2}{\omega_d ||\mathbf{x} - \rho\mu||^d},$$

where μ is a vector orienting the center of the distribution, ρ is a parameter to control the concentration of the distribution around the vector μ and it is related to the variance of the distribution. Recall that, for $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $||x|| = \sqrt{x_1^2 + \ldots + x_d^2}$. Furthermore, $\omega_d = 2\pi^{d/2}[\Gamma(d/2)]^{-1}$ is the surface area of the unit sphere in $\mathbb{R}^d$ (see Golzy and Markatou, 2020). When $\rho \to 0$, the Poisson kernel-based density tends to the uniform density on the sphere. Connections of the PKBDs to other distributions are discussed in detail in Golzy and Markatou (2020). Here we note that when $d = 2$, PKBDs reduce to the wrapped Cauchy distribution. Additionally, with precise choice of the parameters ρ and μ the two-dimensional PKBD becomes a two-dimensional projected normal distribution. However, the connection with the d-dimensional projected normal distributions does not carry beyond $d = 2$. Golzy and Markatou (2020) proposed an acceptance-rejection method for simulating data from a PKBD using von Mises-Fisher envelopes (`rejvmf` method). Furthermore Sablica, Hornik and Leydold (2023) proposed new ways for simulating from the PKBD, using angular central Gaussian envelopes (`rejacg`) or using the projected Saw distributions (`rejpsaw`).

Usage

```
dpkb(x, mu, rho, logdens = FALSE)

rpkb(
  n,
  mu,
  rho,
  method = "rejacg",
  tol.eps = .Machine$double.eps^0.25,
  max.iter = 1000
)
```

Arguments

`x`	$n \times d$-matrix (or data.frame) of n data point on the sphere $\mathcal{S}^{d-1}$, with $d \geq 2$.
`mu`	location vector parameter with length indicating the dimension of generated points.
`rho`	Concentration parameter, with $0 \leq$ `rho` < 1.
`logdens`	Logical; if 'TRUE', densities are returned in logarithmic scale.
`n`	number of observations.
`method`	string that indicates the method used for sampling observations. The available methods are • `'rejvmf'` acceptance-rejection algorithm using von Mises-Fisher envelopes (Algorithm in Table 2 of Golzy and Markatou 2020); • `'rejacg'` using angular central Gaussian envelopes (Algorithm in Table 1 of Sablica et al. 2023); • `'rejpsaw'` using projected Saw distributions (Algorithm in Table 2 of Sablica et al. 2023).
`tol.eps`	the desired accuracy of convergence tolerance (for 'rejacg' method).
`max.iter`	the maximum number of iterations (for 'rejacg' method).

Details

This function `dpkb()` computes the density value for a given point `x` from the Poisson kernel-based distribution with mean direction vector `mu` and concentration parameter `rho`.

The number of observations generated is determined by `n` for `rpkb()`. This function returns the $(n \times d)$-matrix of generated n observations on $\mathcal{S}^{(d-1)}$.

A limitation of the `rejvmf` is that the method does not ensure the computational feasibility of the sampler for ρ approaching 1.

If the chosen method is 'rejacg', the function `uniroot`, from the `stat` package, is used to estimate the beta parameter. In this case, the complete results are provided as output.

Value

`dpkb` gives the density value; `rpkb` generates random observations from the PKBD.

Note

If the required packages (`movMF` for `rejvmf` method, and `Tinflex` for `rejpsaw`) are not installed, the function will display a message asking the user to install the missing package(s).

References

Golzy, M. and Markatou, M. (2020) Poisson Kernel-Based Clustering on the Sphere: Convergence Properties, Identifiability, and a Method of Sampling, Journal of Computational and Graphical Statistics, 29:4, 758-770, DOI: 10.1080/10618600.2020.1740713.

Sablica L., Hornik K. and Leydold J. (2023) "Efficient sampling from the PKBD distribution", Electronic Journal of Statistics, 17(2), 2180-2209.

Examples

```
# Generate some data from pkbd density
pkbd_dat <- rpkb(10, c(0.5, 0), 0.5)

# Calculate the PKBD density values
dens_val <- dpkb(pkbd_dat, c(0.5, 0.5), 0.5)
```

`kb.test` *Kernel-based quadratic distance (KBQD) Goodness-of-Fit tests*

Description

This function performs the kernel-based quadratic distance goodness-of-fit tests. It includes tests for multivariate normality, two-sample tests and k-sample tests.

Usage

```
kb.test(
  x,
  y = NULL,
  h = NULL,
  method = "subsampling",
  B = 150,
  b = NULL,
  Quantile = 0.95,
  mu_hat = NULL,
  Sigma_hat = NULL,
  centeringType = "Nonparam",
  K_threshold = 10,
  alternative = "skewness"
)

## S4 method for signature 'ANY'
```

```
kb.test(
  x,
  y = NULL,
  h = NULL,
  method = "subsampling",
  B = 150,
  b = 0.9,
  Quantile = 0.95,
  mu_hat = NULL,
  Sigma_hat = NULL,
  centeringType = "Nonparam",
  K_threshold = 10,
  alternative = "skewness"
)

## S4 method for signature 'kb.test'
show(object)
```

Arguments

`x`	Numeric matrix or vector of data values.
`y`	Numeric matrix or vector of data values. Depending on the input `y`, the corresponding test is performed. • if `y` = NULL, the function performs the tests for normality on `x` • if `y` is a data matrix, with same dimensions of `x`, the function performs the two-sample test between `x` and `y`. • if `y` is a numeric or factor vector, indicating the group memberships for each observation, the function performs the k-sample test.
`h`	Bandwidth for the kernel function. If a value is not provided, the algorithm for the selection of an optimal h is performed automatically. See the function `select_h` for more details.
`method`	The method used for critical value estimation ("subsampling", "bootstrap", or "permutation")(default: "subsampling").
`B`	The number of iterations to use for critical value estimation (default: 150).
`b`	The size of the subsamples used in the subsampling algorithm (default: 0.8).
`Quantile`	The quantile to use for critical value estimation, 0.95 is the default value.
`mu_hat`	Mean vector for the reference distribution.
`Sigma_hat`	Covariance matrix of the reference distribution.
`centeringType`	String indicating the method used for centering the normal kernel ('Param' or 'Nonparam').
`K_threshold`	maximum number of groups allowed. Default is 10. It is a control parameter. Change in case of more than 10 samples.
`alternative`	Family of alternative chosen for selecting h, between "location", "scale" and "skewness" (only if `h` is not provided).
`object`	Object of class `kb.test`

Details

The function `kb.test` performs the kernel-based quadratic distance tests using the Gaussian kernel with bandwidth parameter `h`. Depending on the shape of the input `y` the function performs the tests of multivariate normality, the non-parametric two-sample tests or the k-sample tests.

The quadratic distance between two probability distributions F and G is defined as

$$d_K(F,G) = \iint K(x,y)d(F-G)(x)d(F-G)(y),$$

where G is a distribution whose goodness of fit we wish to assess and K denotes the Normal kernel defined as

$$K_h(\mathbf{s},\mathbf{t}) = (2\pi)^{-d/2} \left(\det \mathbf{\Sigma}_h\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{s}-\mathbf{t})^\top \mathbf{\Sigma}_h^{-1}(\mathbf{s}-\mathbf{t})\right\},$$

for every $\mathbf{s},\mathbf{t} \in \mathbb{R}^d \times \mathbb{R}^d$, with covariance matrix $\mathbf{\Sigma}_h = h^2 I$ and tuning parameter h.

- **Test for Normality**:
  Let $x_1, x_2, ..., x_n$ be a random sample with empirical distribution function $\hat{F}$. We test the null hypothesis of normality, i.e. $H_0 : F = G = \mathcal{N}_d(\mu, \Sigma)$.
  We consider the U-statistic estimate of the sample KBQD

  $$U_n = \frac{1}{n(n-1)} \sum_{i=2}^{n} \sum_{j=1}^{i-1} K_{cen}(\mathbf{x}_i, \mathbf{x}_j),$$

  then the first test statistics is

  $$T_n = \frac{U_n}{\sqrt{Var(U_n)}},$$

  with $Var(U_n)$ computed exactly following Lindsay et al.(2014), and the V-statistic estimate

  $$V_n = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} K_{cen}(\mathbf{x}_i, \mathbf{x}_j),$$

  where K_{cen} denotes the Normal kernel K_h with parametric centering with respect to the considered normal distribution $G = \mathcal{N}_d(\mu, \Sigma)$.
  The asymptotic distribution of the V-statistic is an infinite combination of weighted independent chi-squared random variables with one degree of freedom. The cutoff value is obtained using the Satterthwaite approximation $c \cdot \chi^2_{DOF}$, where c and DOF are computed exactly following the formulas in Lindsay et al.(2014).
  For the U-statistic the cutoff is determined empirically:
  - Generate data from the considered normal distribution ;
  - Compute the test statistics for `B` Monte Carlo(MC) replications;
  - Compute the 95th quantile of the empirical distribution of the test statistic.
- **k-sample test**:
  Consider k random samples of i.i.d. observations $\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)} \sim F_i$, $i = 1, \ldots, k$. We test if the samples are generated from the same *unknown* distribution, that is $H_0 : F_1 = F_2 =$

$\ldots = F_k$ versus $H_1 : F_i \neq F_j$, for some $1 \leq i \neq j \leq k$.
We construct a matrix distance $\hat{\mathbf{D}}$, with off-diagonal elements

$$\hat{D}_{ij} = \frac{1}{n_i n_j} \sum_{\ell=1}^{n_i} \sum_{r=1}^{n_j} K_{\bar{F}}(\mathbf{x}_\ell^{(i)}, \mathbf{x}_r^{(j)}), \qquad \text{for } i \neq j$$

and in the diagonal

$$\hat{D}_{ii} = \frac{1}{n_i(n_i - 1)} \sum_{\ell=1}^{n_i} \sum_{r\neq\ell}^{n_i} K_{\bar{F}}(\mathbf{x}_\ell^{(i)}, \mathbf{x}_r^{(i)}), \qquad \text{for } i = j,$$

where $K_{\bar{F}}$ denotes the Normal kernel K_h centered non-parametrically with respect to

$$\bar{F} = \frac{n_1 \hat{F}_1 + \ldots + n_k \hat{F}_k}{n}, \quad \text{with } n = \sum_{i=1}^{k} n_i.$$

We compute the trace statistic

$$\text{trace}(\hat{\mathbf{D}}_n) = \sum_{i=1}^{k} \hat{D}_{ii}$$

and D_n, derived considering all the possible pairwise comparisons in the *k*-sample null hypothesis, given as

$$D_n = (k-1)\text{trace}(\hat{\mathbf{D}}_n) - 2 \sum_{i=1}^{k} \sum_{j>i}^{k} \hat{D}_{ij}.$$

We compute the empirical critical value by employing numerical techniques such as the bootstrap, permutation and subsampling algorithms:

- Generate k-tuples, of total size n_B, from the pooled sample following one of the sampling methods;
- Compute the k-sample test statistic;
- Repeat `B` times;
- Select the 95^{th} quantile of the obtained values.

• **Two-sample test**:
Let $x_1, x_2, ..., x_{n_1} \sim F$ and $y_1, y_2, ..., y_{n_2} \sim G$ be random samples from the distributions F and G, respectively. We test the null hypothesis that the two samples are generated from the same *unknown* distribution, that is $H_0 : F = G$ vs $H_1 : F \neq G$. The test statistics coincide with the k-sample test statistics when $k = 2$.

Kernel centering:

The arguments `mu_hat` and `Sigma_hat` indicate the normal model considered for the normality test, that is $H_0 : F = N($`mu_hat`, `Sigma_hat`$)$. For the two-sample and k-sample tests, `mu_hat` and `Sigma_hat` can be used for the parametric centering of the kernel, in the case we want to specify the reference distribution, with `centeringType = "Param"`. This is the default method when the test for normality is performed. The normal kernel centered with respect to $G \sim N_d(\mu, \mathbf{V})$ can be computed as

$$K_{cen(G)}(\mathbf{s}, \mathbf{t}) = K_{\mathbf{\Sigma_h}}(\mathbf{s}, \mathbf{t}) - K_{\mathbf{\Sigma_h}+\mathbf{V}}(\mu, \mathbf{t}) - K_{\mathbf{\Sigma_h}+\mathbf{V}}(\mathbf{s}, \mu) + K_{\mathbf{\Sigma_h}+2\mathbf{V}}(\mu, \mu).$$

We consider the non-parametric centering of the kernel with respect to $\bar{F} = (n_1 F_1 + \dots n_k F_k)/n$ where $n = \sum_{i=1}^{k} n_i$, with `centeringType = "Nonparam"`, for the two- and k-sample tests. Let $\mathbf{z}_1, \dots, \mathbf{z}_n$ denote the pooled sample. For any $s, t \in \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$, it is given by

$$K_{cen(\bar{F})}(\mathbf{s}, \mathbf{t}) = K(\mathbf{s}, \mathbf{t}) - \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{s}, \mathbf{z}_i) - \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{z}_i, \mathbf{t}) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} K(\mathbf{z}_i, \mathbf{z}_j).$$

Value

An S4 object of class `kb.test` containing the results of the kernel-based quadratic distance tests, based on the normal kernel. The object contains the following slots:

- `method`: Description of the kernel-based quadratic distance test performed.
- `x` Data list of samples X (and Y).
- `Un` The value of the U-statistic.
- `H0_Un` A logical value indicating whether or not the null hypothesis is rejected according to Un.
- `CV_Un` The critical value computed for the test Un.
- `Vn` The value of the V-statistic (if available).
- `H0_Vn` A logical value indicating whether or not the null hypothesis is rejected according to Vn (if available).
- `CV_Vn` The critical value computed for the test Vn (if available).
- `h` List with the value of bandwidth parameter used for the normal kernel function. If `select_h` is used, the matrix of computed power values and the corresponding power plot are also provided.
- `B` Number of bootstrap/permutation/subsampling replications.
- `var_Un` exact variance of the kernel-based U-statistic.
- `cv_method` The method used to estimate the critical value (one of "subsampling", "permutation" or "bootstrap").

Note

For the two- and k-sample tests, the slots `Vn`, `H0_Vn` and `CV_Vn` are empty, while the computed statistics are both reported in slots `Un`, `H0_Un` and `CV_Un`.

A U-statistic is a type of statistic that is used to estimate a population parameter. It is based on the idea of averaging over all possible *distinct* combinations of a fixed size from a sample. A V-statistic considers all possible tuples of a certain size, not just distinct combinations and can be used in contexts where unbiasedness is not required.

References

Markatou, M. and Saraceno, G. (2024). “A Unified Framework for Multivariate Two- and k-Sample Kernel-based Quadratic Distance Goodness-of-Fit Tests.”
https://doi.org/10.48550/arXiv.2407.16374

Lindsay, B.G., Markatou, M. and Ray, S. (2014) "Kernels, Degrees of Freedom, and Power Properties of Quadratic Distance Goodness-of-Fit Tests", Journal of the American Statistical Association, 109:505, 395-410, DOI: 10.1080/01621459.2013.836972

See Also

kb.test for the class definition.

Examples

```
# create a kb.test object
x <- matrix(rnorm(100), ncol = 2)
y <- matrix(rnorm(100), ncol = 2)

# Normality test
my_test <- kb.test(x, h=0.5)
my_test

# Two-sample test
my_test <- kb.test(x, y, h = 0.5, method = "subsampling", b = 0.9,
                   centeringType = "Nonparam")
my_test

# k-sample test
z <- matrix(rnorm(100, 2), ncol = 2)
dat <- rbind(x, y, z)
group <- rep(c(1, 2, 3), each = 50)
my_test <- kb.test(x = dat, y = group, h = 0.5, method = "subsampling", b = 0.9)
my_test
```

`kb.test-class` *An S4 class for kernel-based distance tests with normal kernel*

Description

A class to represent the results of Gaussian kernel-based quadratic distance tests. This includes the normality test, the two-sample test statistics and the k-sample tests.

Slots

`method` String indicating the kernel-based quadratic distance test performed.

`Un` The value of the test U-statistic.

`Vn` The value of the test V-statistic.

`H0_Un` A logical value indicating whether or not the null hypothesis is rejected according to U-statistic.

`H0_Vn` A logical value indicating whether or not the null hypothesis is rejected according to Vn.

`data` List of samples X (and Y).

`CV_Un` The critical value computed for the test Un.

`CV_Vn` The critical value computed for the test Vn.

cv_method The method used to estimate the critical value (one of "subsampling", "permutation" or "bootstrap").

h A list with the value of bandwidth parameter used for the Gaussian kernel. If the function select_h is used, then also the matrix of computed power values and the resulting power plot are provided.

B Number of bootstrap/permutation/subsampling replications.

var_Un Exact variance of the kernel-based U-statistic.

See Also

kb.test() for the function that generates this class.

Examples

```
# create a kb.test object
x <- matrix(rnorm(100), ncol = 2)
y <- matrix(rnorm(100), ncol = 2)
# Normality test
kb.test(x, h = 0.5)

# Two-sample test
kb.test(x, y, h=0.5, method = "subsampling", b = 0.9)
```

---

pk.test *Poisson kernel-based quadratic distance test of Uniformity on the sphere*

Description

This function performs the kernel-based quadratic distance goodness-of-fit tests for Uniformity for multivariate spherical data x on $\mathcal{S}^{d-1}$ using the Poisson kernel with concentration parameter rho. The Poisson kernel-based test for uniformity exhibits excellent results especially in the case of multimodal distributions, as shown in the example of the Uniformity test on the Sphere vignette.

Usage

```
pk.test(x, rho, B = 300, Quantile = 0.95)

## S4 method for signature 'ANY'
pk.test(x, rho, B = 300, Quantile = 0.95)

## S4 method for signature 'pk.test'
show(object)
```

Arguments

`x`	A numeric $(n \times d)$-matrix of n data points on the Sphere $\mathcal{S}^(d-1)$ as rows.
`rho`	Concentration parameter of the Poisson kernel function.
`B`	Number of Monte Carlo iterations for critical value estimation of Un (default: 300).
`Quantile`	The quantile to use for critical value estimation, 0.95 is the default value.
`object`	Object of class `pk.test`

Details

Let $x_1, x_2, ..., x_n$ be a random sample with empirical distribution function $\hat{F}$. We test the null hypothesis of uniformity on the $(d-1)$-dimensional sphere, i.e. $H_0 : F = G$, where G is the uniform distribution on the $(d-1)$-dimensional sphere $\mathcal{S}^{d-1}$. We compute the U-statistic estimate of the sample KBQD (Kernel-Based Quadratic Distance)

$$U_n = \frac{1}{n(n-1)} \sum_{i=2}^{n} \sum_{j=1}^{i-1} K_{cen}(\mathbf{x}_i, \mathbf{x}_j),$$

then the first test statistic is given as

$$T_n = \frac{U_n}{\sqrt{Var(U_n)}},$$

with

$$Var(U_n) = \frac{2}{n(n-1)} \left[\frac{1+\rho^2}{(1-\rho^2)^{d-1}} - 1 \right],$$

and the V-statistic estimate of the KBQD

$$V_n = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} K_{cen}(\mathbf{x}_i, \mathbf{x}_j),$$

where K_{cen} denotes the Poisson kernel K_ρ centered with respect to the uniform distribution on the $(d-1)$-dimensional sphere, that is

$$K_{cen}(\mathbf{u}, \mathbf{v}) = K_\rho(\mathbf{u}, \mathbf{v}) - 1$$

and

$$K_\rho(\mathbf{u}, \mathbf{v}) = \frac{1-\rho^2}{(1+\rho^2 - 2\rho(\mathbf{u} \cdot \mathbf{v}))^{d/2}},$$

for every $\mathbf{u}, \mathbf{v} \in \mathcal{S}^{d-1} \times \mathcal{S}^{d-1}$.

The asymptotic distribution of the V-statistic is an infinite combination of weighted independent chi-squared random variables with one degree of freedom. The cutoff value is obtained using the Satterthwaite approximation $c \cdot \chi^2_{DOF}$, where

$$c = \frac{(1+\rho^2) - (1-\rho^2)^{d-1}}{(1+\rho)^d - (1-\rho^2)^{d-1}}$$

and

$$DOF(K_{cen}) = \left(\frac{1+\rho}{1-\rho}\right)^{d-1} \left\{ \frac{\left(1+\rho-(1-\rho)^{d-1}\right)^2}{1+\rho^2-(1-\rho^2)^{d-1}} \right\}.$$

. For the U-statistic the cutoff is determined empirically:

- Generate data from a Uniform distribution on the d-dimensional sphere;
- Compute the test statistics for `B` Monte Carlo(MC) replications;
- Compute the 95th quantile of the empirical distribution of the test statistic.

Value

An S4 object of class `pk.test` containing the results of the Poisson kernel-based tests. The object contains the following slots:

- `method`: Description of the test performed.
- `x` Data matrix.
- `Un` The value of the U-statistic.
- `CV_Un` The empirical critical value for Un.
- `H0_Vn` A logical value indicating whether or not the null hypothesis is rejected according to Un.
- `Vn` The value of the V-statistic Vn.
- `CV_Vn` The critical value for Vn computed following the asymptotic distribution.
- `H0_Vn` A logical value indicating whether or not the null hypothesis is rejected according to Vn.
- `rho` The value of concentration parameter used for the Poisson kernel function.
- `B` Number of replications for the critical value of the U-statistic Un.

Note

A U-statistic is a type of statistic that is used to estimate a population parameter. It is based on the idea of averaging over all possible *distinct* combinations of a fixed size from a sample. A V-statistic considers all possible tuples of a certain size, not just distinct combinations and can be used in contexts where unbiasedness is not required.

References

Ding, Y., Markatou, M. and Saraceno, G. (2023). “Poisson Kernel-Based Tests for Uniformity on the d-Dimensional Sphere.” Statistica Sinica. doi:10.5705/ss.202022.0347

See Also

pk.test

Examples

```
# create a pk.test object
x_sp <- sample_hypersphere(3, n_points = 100)
unif_test <- pk.test(x_sp, rho = 0.8)
unif_test
```

---

pk.test-class *An S4 class for Poisson kernel-based quadratic distance tests.*

Description

A class to represent the results of Poisson kernel-based quadratic distance tests for Uniformity on the sphere.

Slots

`method` Description of the test.

`x` Matrix of data

`Un` The value of the U-statistic.

`CV_Un` The critical value for Un computed through replications.

`H0_Un` A logical value indicating whether or not the null hypothesis is rejected according to Un.

`Vn` The value of the V-statistic.

`CV_Vn` The critical value for Vn computed following the asymptotic distribution.

`H0_Vn` A logical value indicating whether or not the null hypothesis is rejected according to Vn.

`rho` The concentration parameter of the Poisson kernel.

`B` Number of replications.

`var_Un` exact variance of the kernel-based U-statistic.

See Also

`pk.test()`

Examples

```
# create a pk.test object
d=3
size=100
x_sp <- sample_hypersphere(d, n_points=size)
pk.test(x_sp,rho=0.8)
```

pkbc *Poisson kernel-based clustering on the sphere*

Description

The function `pkbc()` performs the Poisson kernel-based clustering algorithm on the sphere proposed by Golzy and Markatou (2020). The proposed algorithm is based on a mixture, with M components, of Poisson kernel-based densities on the hypersphere $\mathcal{S}^{d-1}$ given by

$$f(x|\Theta) = \sum_{j=1}^{M} \alpha_j f_j(x|\rho_j, \mu_j)$$

where α_j's are the mixing proportions and $f_j(x|\rho_j, \mu_j)$'s denote the probability density function of a d-variate Poisson kernel-based density given as

$$f(\mathbf{x}|\rho, \mu) = \frac{1-\rho^2}{\omega_d ||\mathbf{x} - \rho\mu||^d}.$$

The parameters α_j, μ_j, ρ_j are estimated through a iterative reweighted EM algorithm.
The proposed clustering algorithm exhibits excellent results when (1) the clusters are not well separated; (2) the data points are fairly well concentrated around the vectors μ_j of each cluster; (3) the percentage of noise in the data increases.

Usage

```
pkbc(
  dat,
  nClust,
  maxIter = 300,
  stoppingRule = "loglik",
  initMethod = "sampleData",
  numInit = 10
)

## S4 method for signature 'ANY'
pkbc(
  dat,
  nClust,
  maxIter = 300,
  stoppingRule = "loglik",
  initMethod = "sampleData",
  numInit = 10
)

## S4 method for signature 'pkbc'
show(object)
```

Arguments

`dat`	$(n \times d)$-data matrix or data.frame of data points on the sphere to be clustered. The observations in `dat` are normalized by dividing with the length of the vector to ensure that they lie on the d-dimensional sphere. Note that $d > 1$.
`nClust`	Number of clusters. It can be a single value or a numeric vector.
`maxIter`	The maximum number of iterations before a run is terminated.
`stoppingRule`	String describing the stopping rule to be used within each run. Currently must be either `'max'`, `'membership'`, or `'loglik'`.
`initMethod`	String describing the initialization method to be used. Currently must be `'sampleData'`.
`numInit`	Number of initialization.
`object`	Object of class `pkbc`

Details

We set all concentration parameters equal to 0.5 and all mixing proportions to be equal.
The initialization method `'sampleData'` indicates that observation points are randomly chosen as initializers of the centroids μ_j. This random starts strategy has a chance of not obtaining initial representatives from the underlying clusters, then the clustering is performed `numInit` times and the random start with the highest likelihood is chosen as the final estimate of the parameters.

The possible `stoppingRule` for each iteration are:

- `'loglik'` run the algorithm until the change in log-likelihood from one iteration to the next is less than a given threshold (1e-7)
- `'membership'` run the algorithm until the membership is unchanged for all points from one iteration to the next
- `'max'` reach a maximum number of iterations `maxIter`

The obtained estimates are used for assigning final memberships, identifying the `nClust` clusters, according to the following rule

$$P(x_i, \Theta) = \arg \max_{j \in \{1,\ldots,k\}} \{ \frac{\alpha_j f_j(x_i|\mu_j, \rho_j)}{f(x_i, \Theta)} \}.$$

The number of clusters `nClust` must be provided as input to the clustering algorithm.

Value

An S4 object of class `pkbc` containing the results of the clustering procedure based on Poisson kernel-based distributions. The object contains the following slots:

`res_k`: List of results of the Poisson kernel-based clustering algorithm for each value of number of clusters specified in `nClust`. Each object in the list contains:

- `postProbs` Posterior probabilities of each observation for the indicated clusters.
- `LogLik` Maximum value of log-likelihood function

- `wcss` Values of within-cluster sum of squares computed with Euclidean distance and cosine similarity, respectively.
- `params` List of estimated parameters of the mixture model
  - `mu` estimated centroids
  - `rho` estimated concentration parameters rho
  - `alpha` estimated mixing proportions
- `finalMemb` Vector of final memberships
- `runInfo` List of information of the EM algorithm iterations
  - `lokLikVec` vector of log-likelihood values
  - `numIterPerRun` number of E-M iterations per run

`input`: List of input information.

Note

The clustering algorithm is tailored for data points on the sphere $\mathcal{S}^{d-1}$, but it can also be performed on spherically transformed observations, i.e. data points on the Euclidean space $\mathbb{R}^d$ that are normalized such that they lie on the corresponding $(d-1)$-dimensional sphere $\mathcal{S}^{d-1}$.

References

Golzy, M. and Markatou, M. (2020) Poisson Kernel-Based Clustering on the Sphere: Convergence Properties, Identifiability, and a Method of Sampling, Journal of Computational and Graphical Statistics, 29:4, 758-770, DOI: 10.1080/10618600.2020.1740713.

See Also

`dpkb()` and `rpkb()` for more information on the Poisson kernel-based distribution.
pkbc for the class definition.

Examples

```
# We generate three samples of 100 observations from 3-dimensional
# Poisson kernel-based densities with rho=0.8 and different mean directions
size <- 100
groups <- c(rep(1, size), rep(2, size), rep(3, size))
rho <- 0.8
set.seed(081423)
data1 <- rpkb(size, c(1, 0, 0), rho)
data2 <- rpkb(size, c(0, 1, 0), rho)
data3 <- rpkb(size, c(0, 0, 1), rho)
dat <- rbind(data1, data2, data3)

# Perform the clustering algorithm with number of clusters k=3.
pkbd <- pkbc(dat = dat, nClust = 3)
show(pkbd)
```

pkbc-class *A S4 class for the clustering algorithm on the sphere based on Poisson kernel-based distributions.*

Description

A class to represent the results of Poisson kernel-based clustering procedure for spherical observations.

Slots

res_k List of objects with the results of the clustering algorithm for each value of possible number of clusters considered.

input List of input data

See Also

pkbc() for more details.

Examples

```
data("wireless")
res <- pkbc(as.matrix(wireless[,-8]),4)
```

pkbc_validation *Validation of Poisson kernel-based clustering results*

Description

Method for objects of class pkbc which computes evaluation measures for clustering results. The following evaluation measures are computed: In-Group Proportion (Kapp and Tibshirani (2007)). If true label are provided, ARI, Average Silhouette Width (Rousseeuw (1987)), Macro-Precision and Macro-Recall are computed.

Usage

```
pkbc_validation(object, true_label = NULL)
```

Arguments

object	Object of class pkbc
true_label	factor or vector of true membership to clusters (if available). It must have the same length of final memberships.

Details

The IGP is a statistical measure that quantifies the proportion of observations within a group that belong to the same predefined category or class. It is often used to assess the homogeneity of a group by evaluating how many of its members share the same label. A higher IGP indicates that the group is more cohesive, while a lower proportion suggests greater diversity or misclassification within the group (Kapp and Tibshirani 2007).

The Adjusted Rand Index (ARI) is a statistical measure used in data clustering analysis. It quantifies the similarity between two partitions of a dataset by comparing the assignments of data points to clusters. The ARI value ranges from 0 to 1, where a value of 1 indicates a perfect match between the partitions and a value close to 0 indicates a random assignment of data points to clusters.

The average silhouette width quantifies the quality of clustering by measuring how well each object fits within its assigned cluster. It is the mean of silhouette values, which compare the tightness of an object within its cluster to its separation from other clusters. Higher values indicate well-separated, cohesive clusters, making it useful for selecting the *appropriate* number of clusters (Rousseeuw 1987).

Macro Precision is a metric used in multi-class classification that calculates the precision for each class independently and then takes the average of these values. Precision for a class is defined as the proportion of true positive predictions out of all predictions made for that class.

Macro Recall is similar to Macro Precision but focuses on recall. Recall for a class is the proportion of true positive predictions out of all actual instances of that class. Macro Recall is the average of the recall values computed for each class.

Value

List with the following components:

- `metrics` Table of computed evaluation measures for each value of number of clusters in the `pkbc` object. The number of cluster is indicated as column name.
- `IGP` List of in-group proportions for each value of number of clusters specified.

Note

Note that Macro Precision and Macro Recall depend on the assigned labels, while the ARI measures the similarity between partition up to label switching.

If the required packages (`mclust` for ARI, `clusterRepro` for IGP, and `cluster` for ASW) are not installed, the function will display a message asking the user to install the missing package(s).

References

Kapp, A.V. and Tibshirani, R. (2007) "Are clusters found in one dataset present in another dataset?", Biostatistics, 8(1), 9–31, https://doi.org/10.1093/biostatistics/kxj029

Rousseeuw, P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65.

See Also

`pkbc()` for the clustering algorithm
`pkbc` for the class object definition.

Examples

```
#We generate three samples of 100 observations from 3-dimensional
#Poisson kernel-based densities with rho=0.8 and different mean directions

size <- 20
groups <- c(rep(1, size), rep(2, size), rep(3, size))
rho <- 0.8
set.seed(081423)
data1 <- rpkb(size, c(1,0,0), rho, method = 'rejvmf')
data2 <- rpkb(size, c(0,1,0), rho, method = 'rejvmf')
data3 <- rpkb(size, c(1,0,0), rho, method = 'rejvmf')
data <- rbind(data1, data2, data3)

#Perform the clustering algorithm
pkbc_res <- pkbc(data, 3)
pkbc_validation(pkbc_res)
```

---

plot.pkbc *Plotting method for Poisson kernel-based clustering*

Description

Plots for a pkbc object.

Usage

```
## S4 method for signature 'pkbc,ANY'
plot(x, k = NULL, true_label = NULL, pca_res = FALSE, ...)
```

Arguments

`x`	Object of class pkbc
`k`	number of considered clusters. If it is not provided the scatter plot is displayed for each value of number of clusters present in the `x` object
`true_label`	factor or vector of true membership to clusters (if available). It must have the same length of final memberships.
`pca_res`	Logical. If TRUE the results from PCALocantore are also reported (when dimension is greater than 3).
`...`	Additional arguments that can be passed to the plot function

Details

- scatterplot: If dimension is equal to 2 or 3, points are displayed on the circle and sphere, respectively. If dimension if greater than 3, the spherical Principal Component procedure proposed by Locantore et al. (1999), is applied for dimensionality reduction and the first three principal components are normalized and displayed on the sphere. For d > 3, the complete results from the `PcaLocantore` function (package `rrcov`) are returned if `pca_res=TRUE`.
- elbow plot: the within cluster sum of squares (wcss) is computed using the Euclidean distance (left) and the cosine similarity (right).

Value

The scatter-plot(s) and the elbow plot.

Note

The elbow plot is commonly used as a graphical method for choosing the *appropriate* number of clusters. Specifically, plotting the wcss versus the number of clusters, the suggested number of clusters correspond to the point in which the plotted line has the greatest change in slope, showing an elbow.

References

Locantore, N., Marron, J.S., Simpson, D.G. et al. (1999) "Robust principal component analysis for functional data." Test 8, 1–73. https://doi.org/10.1007/BF02595862

See Also

`pkbc()` for the clustering algorithm
`pkbc` for the class object definition.

Examples

```
dat <- matrix(rnorm(300), ncol = 3)
pkbc_res <- pkbc(dat, 3)
plot(pkbc_res, 3)
```

`predict.pkbc` *Cluster spherical observations using a mixture of Poisson kernel-based densities*

Description

Obtain predictions of membership for spherical observations based on a mixture of Poisson kernel-based densities estimated by `pkbc`

Usage

```
## S4 method for signature 'pkbc'
predict(object, k, newdata = NULL)
```

Arguments

`object`	Object of class `pkbc`
`k`	Number of clusters to be used.
`newdata`	a data.frame or a matrix of the data. If missing the clustering data obtained from the `pkbc` object are classified.

Value

Returns a list with the following components

- Memb: vector of predicted memberships of `newdata`
- Probs: matrix where entry (i,j) denotes the probability that observation i belongs to the k-th cluster.

See Also

`pkbc()` for the clustering algorithm
`pkbc` for the class object definition.

Examples

```
# generate data
dat <- rbind(matrix(rnorm(100), ncol = 2), matrix(rnorm(100, 5), ncol = 2))
res <- pkbc(dat, 2)

# extract membership of dat
predict(res, k = 2)
# predict membership of new data
newdat <- rbind(matrix(rnorm(10), ncol = 2), matrix(rnorm(10, 5), ncol = 2))
predict(res, k = 2, newdat)
```

`sample_hypersphere` *Generate random sample from the hypersphere*

Description

Generate a random sample from the uniform distribution on the hypersphere.

Usage

```
sample_hypersphere(d, n_points = 1)
```

Arguments

`d`	Number of dimensions.
`n_points`	Number of sampled observations.

Value

Data matrix with the sampled observations.

Examples

```
x_sp <- sample_hypersphere(3,100)
```

`select_h` *Select the value of the kernel tuning parameter*

Description

This function computes the kernel bandwidth of the Gaussian kernel for the normality, two-sample and k-sample kernel-based quadratic distance (KBQD) tests.

Usage

```
select_h(
  x,
  y = NULL,
  alternative = NULL,
  method = "subsampling",
  b = 0.8,
  B = 100,
  delta_dim = 1,
  delta = NULL,
  h_values = NULL,
  Nrep = 50,
  n_cores = 2,
  Quantile = 0.95,
  power.plot = TRUE
)
```

Arguments

`x` Data set of observations from X.

`y` Numeric matrix or vector of data values. Depending on the input `y`, the selection of h is performed for the corresponding test.

- if `y` = NULL, the function performs the tests for normality on `x`.

- if `y` is a data matrix, with same dimensions of `x`, the function performs the two-sample test between `x` and `y`.
- if `y` is a numeric or factor vector, indicating the group memberships for each observation, the function performs the k-sample test.

`alternative`	Family of alternative chosen for selecting h, between "location", "scale" and "skewness".
`method`	The method used for critical value estimation ("subsampling", "bootstrap", or "permutation").
`b`	The size of the subsamples used in the subsampling algorithm .
`B`	The number of iterations to use for critical value estimation, B = 150 as default.
`delta_dim`	Vector of coefficient of alternative with respect to each dimension
`delta`	Vector of parameter values indicating chosen alternatives
`h_values`	Values of the tuning parameter used for the selection
`Nrep`	Number of bootstrap/permutation/subsampling replications.
`n_cores`	Number of cores used to parallel the h selection algorithm. If this is not provided, the function will detect the available cores.
`Quantile`	The quantile to use for critical value estimation, 0.95 is the default value.
`power.plot`	Logical. If TRUE, it is displayed the plot of power for values in h_values and delta.

Details

The function performs the selection of the optimal value for the tuning parameter h of the normal kernel function, for normality test, the two-sample and k-sample KBQD tests. It performs a small simulation study, generating samples according to the family of `alternative` specified, for the chosen values of `h_values` and `delta`.

We consider target alternatives $F_\delta(\hat{\mu}, \hat{\mathbf{\Sigma}}, \hat{\lambda})$, where $\hat{\mu}, \hat{\mathbf{\Sigma}}$ and $\hat{\lambda}$ indicate the location, covariance and skewness parameter estimates from the pooled sample.

- Compute the estimates of the mean $\hat{\mu}$, covariance matrix $\hat{\Sigma}$ and skewness $\hat{\lambda}$ from the pooled sample.
- Choose the family of alternatives $F_\delta = F_\delta(\hat{\mu}, \hat{\Sigma}, \hat{\lambda})$.

  For each value of δ and h:
- Generate $\mathbf{X}_1, \ldots, \mathbf{X}_{k-1} \sim F_0$, for $\delta = 0$;
- Generate $\mathbf{X}_k \sim F_\delta$;
- Compute the k-sample test statistic between $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ with kernel parameter h;
- Compute the power of the test. If it is greater than 0.5, select h as optimal value.
- If an optimal value has not been selected, choose the h which corresponds to maximum power.

The available `alternative` are
location alternatives, $F_\delta = SN_d(\hat{\mu} + \delta, \hat{\Sigma}, \hat{\lambda})$,with $\delta = 0.2, 0.3, 0.4$;
scale alternatives, $F_\delta = SN_d(\hat{\mu}, \hat{\Sigma} * \delta, \hat{\lambda})$, $\delta = 0.1, 0.3, 0.5$;

skewness alternatives, $F_\delta = SN_d(\hat{\mu}, \hat{\Sigma}, \hat{\lambda} + \delta)$, with $\delta = 0.2, 0.3, 0.6$.
The values of $h = 0.6, 1, 1.4, 1.8, 2.2$ and $N = 50$ are set as default values.
The function `select_h()` allows the user to set the values of δ and h for a more extensive grid search. We suggest to set a more extensive grid search when computational resources permit.

Value

A list with the following attributes:

- `h_sel` the selected value of tuning parameter h;
- `power` matrix of power values computed for the considered values of `delta` and `h_values`;
- `power.plot` power plots (if `power.plot` is `TRUE`).

Note

Please be aware that the `select_h()` function may take a significant amount of time to run, especially with larger datasets or when using an larger number of parameters in `h_values` and `delta`. Consider this when applying the function to large or complex data.

References

Markatou, M. and Saraceno, G. (2024). “A Unified Framework for Multivariate Two- and k-Sample Kernel-based Quadratic Distance Goodness-of-Fit Tests.”
https://doi.org/10.48550/arXiv.2407.16374

Saraceno, G., Markatou, M., Mukhopadhyay, R. and Golzy, M. (2024). Goodness-of-Fit and Clustering of Spherical Data: the QuadratiK package in R and Python.
https://arxiv.org/abs/2402.02290.

See Also

The function `select_h` is used in the `kb.test()` function.

Examples

```
# Select the value of h using the mid-power algorithm

x <- matrix(rnorm(100), ncol = 2)
y <- matrix(rnorm(100), ncol = 2)
h_sel <- select_h(x, y, "skewness")
h_sel
```

`stats_clusters` *Descriptive statistics for the clusters identified by the Poisson kernel-based clustering.*

Description

Method for objects of class pkbc which computes some descriptive for each variable with respect to the detected groups.

Method for objects of class pkbc which computes descriptive statistics for each variable with respect to the detected groups.

Usage

```
stats_clusters(object, ...)

## S4 method for signature 'pkbc'
stats_clusters(object, k)
```

Arguments

`object`	Object of class pkbc.
`...`	possible additional inputs
`k`	Number of clusters to be used.

Details

The function computes mean, standard deviation, median, inter-quantile range, minimum and maximum for each variable in the data set given the final membership assigned by the clustering algorithm.

Value

List with computed descriptive statistics for each dimension.

See Also

`pkbc()` for the clustering algorithm
pkbc for the class object definition.

Examples

```
#We generate three samples of 100 observations from 3-dimensional
#Poisson kernel-based densities with rho=0.8 and different mean directions
dat<-matrix(rnorm(300),ncol=3)

#Perform the clustering algorithm
pkbc_res<- pkbc(dat, 3)
```

```
stats_clusters(pkbc_res, 3)
```

---

summary.kb.test *Summarizing kernel-based quadratic distance results*

Description

`summary` method for the class `kb.test`

Usage

```
## S4 method for signature 'kb.test'
summary(object)
```

Arguments

`object`	Object of class `kb.test`

Value

List with the following components:

- `summary_tables` Table of computed descriptive statistics per variable (and per group if available).
- `test_results` Data frame with the results of the performed kernel-based quadratic distance test.
- `qqplots` Figure with qq-plots for each variable.

See Also

`kb.test()` and kb.test for more details.

Examples

```
# create a kb.test object
x <- matrix(rnorm(100),ncol=2)
# Normality test
my_test <- kb.test(x, h=0.5)
summary(my_test)
```

`summary.pk.test` *Summarizing kernel-based quadratic distance results*

Description

`summary` method for the class `pk.test`

Usage

```
## S4 method for signature 'pk.test'
summary(object)
```

Arguments

`object`	Object of class `pk.test`

Value

List with the following components:

- `summary_tables` Table of computed descriptive statistics per variable.
- `test_results` Data frame with the results of the performed Poisson kernel-based test.
- `qqplots` Figure with qq-plots for each variable against the uniform distribution.

See Also

`pk.test()` and pk.test for additional details.

Examples

```
# create a pk.test object
x_sp <- sample_hypersphere(3, n_points=100)
unif_test <- pk.test(x_sp,rho=0.8)
summary(unif_test)
```

`summary.pkbc` *Summarizing PKBD mixture Fits*

Description

Summary method for class "pkbc"

Usage

```
## S4 method for signature 'pkbc'
summary(object)
```

Arguments

object	Object of class pkbc

Value

Display the logLikelihood values and within cluster sum of squares (wcss) for all the values of number of clusters provided. For each of these values the estimated mixing proportions are showed together with a table with the assigned memberships.

See Also

pkbc() for the clustering algorithm
pkbc for the class object definition.

Examples

```
dat <- rbind(matrix(rnorm(100), 2), matrix(rnorm(100, 5), 2))
res <- pkbc(dat, 2:4)
summary(res)
```

---

wine *Wine data set*

Description

The wine data frame has 178 rows and 14 columns. The first 13 variables report 13 constituents found in each of the three types of wines. The last column indicates the class labels (1,2 or 3).

Usage

```
wine
```

Format

A data frame containing the following columns:

- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins

- `Color intensity`
- `Hue`
- `OD280/OD315 of diluted wines`
- `Proline`
- `y`: class membership

Details

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Source

Aeberhard, S. and Forina, M. (1991). Wine. UCI Machine Learning Repository. https://doi.org/10.24432/C5PC7J.

References

Aeberhard, S., Coomans, D. and De Vel, O. (1994). Comparative analysis of statistical pattern recognition methods in high dimensional settings. Pattern Recognition, 27(8), 1065-1077.

Examples

```
data(wine)
summary(wine)
```

`wireless` *Wireless Indoor Localization*

Description

The `wireless` data frame has 2000 rows and 8 columns. The first 7 variables report the measurements of the Wi-Fi signal strength received from 7 Wi-Fi routers in an office location in Pittsburgh (USA). The last column indicates the class labels.

Usage

```
wireless
```

Format

A data frame containing the following columns:

- `V1` Signal strength from router 1.
- `V2` Signal strength from router 2.
- `V3` Signal strength from router 3.
- `V4` Signal strength from router 4.
- `V5` Signal strength from router 5.
- `V6` Signal strength from router 6.
- `V7` Signal strength from router 7.
- `V8` Group memberships, from 1 to 4.

Details

The Wi-Fi signal strength is measured in dBm, decibel milliwatts, which is expressed as a negative value ranging from -100 to 0. The labels correspond to 4 different rooms. In total, we have 4 groups with 500 observations each.

Source

Bhatt, R. (2017). Wireless Indoor Localization. UCI Machine Learning Repository.
https://doi.org/10.24432/C51880.

References

Rohra, J.G., Perumal, B., Narayanan, S.J., Thakur, P. and Bhatt, R.B. (2017). "User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks". In: Deep, K., et al. Proceedings of Sixth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, vol 546. Springer, Singapore. https://doi.org/10.1007/978-981-10-3322-3_27

Examples

```
data(wireless)
summary(wireless)
```

Index