Package 'CircularSilhouette'

October 12, 2022

Type Package

Title Fast Silhouette on Circular or Linear Data Clusters

Version 0.0.1

Date 2022-04-22

Author Yinong Chen [aut] (<https://orcid.org/0000-0003-1641-1712>),
Tathagata Debnath [aut] (<https://orcid.org/0000-0001-6445-275X>),
Andrew Cai [aut],
Joe Song [aut, cre] (<https://orcid.org/0000-0002-6883-6547>)

Maintainer Joe Song <joemsong@cs.nmsu.edu>

Description Calculating silhouette information for clusters on circular or linear data using fast algorithms. These algorithms run in linear time on sorted data, in contrast to quadratic time by the definition of silhouette. When used together with the fast and optimal circular clustering method FOCC (Debnath & Song 2021) <doi:10.1109/TCBB.2021.3077573> implemented in R package 'OptCirClust', circular silhouette can be maximized to find the optimal number of circular clusters; it can also be used to estimate the period of noisy periodical data.

VignetteBuilder knitr

License LGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Imports OptCirClust, Rcpp (>= 1.0.7), Rdpack

Suggests cluster, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), graphics

LinkingTo Rcpp

NeedsCompilation yes

RdMacros Rdpack

Config/testthat/edition 3

Repository CRAN

Date/Publication 2022-04-27 07:40:02 UTC

R topics documented:

`circular.sil` *Calculating Silhouette on Circular Data Clusters*

Description

A fast linear-time algorithm to calculate silhouette information on circular data with cluster labels.

Usage

```
circular.sil(O, cluster, Circumference, method = c("linear", "quadratic"))
```

Arguments

`O`	a numeric vector of circular data points
`cluster`	an integer vector of cluster labels for each point
`Circumference`	a numeric value giving the circumference of the circle
`method`	a character value to specify the algorithm to calculate the silhouette information. The default value is `"linear"`, indicating a fast linear time algorithm for calculating circular silhouette. The option of `"quadratic"` is provided for testing and comparison, not meant for production use.

Details

If `method` takes the value of `"linear"` (default), the silhouette information on circular data is calculated by a fast linear-time algorithm; if `method` is `"quadratic"`, a quadratic-time algorithm is used instead to calculate silhouette by definition. There is an overhead of sorting $O(n \log n)$ if the input data are not sorted.

One important assumption is that a cluster cannot be contained in another cluster in the input cluster labels.

Value

The function returns a numeric value of the average silhouette information calculated on the input circular data clusters.

Examples

```
O <- c(-1.2, -2, -3, -2.5, 1, 0.8, 1.5, 1.2)
cluster <- c(1, 1, 1, 1, 2, 2, 2, 2)
circular.sil(O, cluster, 3)
```

`estimate.period` *Estimating the Period of Noisy Periodical Data*

Description

By performing circular clustering and calculating circular silhouette, the function estimates the period of periodical data.

Usage

```
estimate.period(x, possible.periods = diff(range(x))/2^(1:5), ks = 2:10)
```

Arguments

`x`	a numeric vector of data points that are one-dimensional, noisy, periodical
`possible.periods`	a numeric vector representing a set of period values to evaluate
`ks`	a numeric vector of numbers of clusters within one period

Details

The user can estimate a period by providing the number of clusters within one period and a set of periods for examination. An optimal circular clustering algorithm `CirClust` in R package **OptCirClust** is used to cluster the periodical data. The algorithm converts the periodical data to circular data of a circumference equal to twice the tested period. Then circular silhouette information for each circumference and number of clusters are computed to find the maximum silhouette information. The half of circumference giving maximum silhouette information is selected to be the estimated period.

The possible periods provided by the function should be close to the true period. This is not ideal and we are improving the design to be more robust.

Value

The function returns a numeric value representing the estimated period.

Examples

```
library(OptCirClust)
x=c(40,41,42,50,51,52,60,61,62,70,71,72,80,81,82,90,91,92)
x <- x + rnorm(length(x))
clusterrange=c(2:5)
periodrange=c(80:120)/10
period<-estimate.period(x, periodrange, clusterrange)
cat("The estimated period is", period, "\n")
plot(x, rep(1, length(x)), type="h", col="purple",
     ylab="", xlab="Noisy periodic data",
     main="Period estimation",
     sub=paste("Estimated period =", period))
k <- (max(x) - min(x)) %/% period
abline(v=min(x)+period/2 + period * (0:k), lty="dashed", col="green3")
```

`fast.sil` *Calculating Silhouette on Linear Data Clusters*

Description

A fast linear-time algorithm to calculate silhouette information on one-dimensional data with cluster labels.

Usage

```
fast.sil(x, cluster)
```

Arguments

`x`	a numeric vector of one-dimensional points
`cluster`	an integer vector of cluster labels for each point

Details

The silhouette information on one-dimensional data is calculated in linear time here, instead of quadratic time by definition. There is an overhead of sorting $O(n \log n)$ if the input data are not sorted.

Value

The function returns a numeric value of the average silhouette information calculated on the input data clusters.

Examples

```
x <- c(-1.2, -2, -3, -2.5, 1, 0.8, 1.5, 1.2)
cluster <- c(1, 1, 1, 1, 2, 2, 2, 2)
fast.sil(x, cluster)
```

`find.num.of.clusters` *Finding an Optimal Number of Circular Data Clusters*

Description

An optimal number of clusters is selected on circular data such that the number maximizes the circular silhouette information.

Usage

```
find.num.of.clusters(O, Circumference, ks = 2:10)
```

Arguments

`O` a numeric vector of coordinates of data points along a circle.

`Circumference` a numeric value giving the circumference of the circle

`ks` an integer vector representing possible choices for the number of clusters

Details

Using the circular clustering algorithm in the R package **OptCirClust** (Debnath and Song 2021), we will examine every value of k in the given choices of number of clusters. We select a k that maximizes the circular silhouette information.

Value

The function returns an integer number that is optimal in maximizing circular silhouette.

References

Debnath T, Song M (2021). "Fast optimal circular clustering and applications on round genomes." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. doi: 10.1109/TCBB.2021.3077573.

Examples

```
library(OptCirClust)
Circumference=100
O=c(99,0,1,2,3,15,16,17,20,50,55,53,70,72,73,69)
K_range=c(2:8)
k <- find.num.of.clusters(O, Circumference, K_range)
result_FOCC <- CirClust(O, k, Circumference, method = "FOCC")
opar <- par(mar=c(0,0,2,0))
plot(result_FOCC, cex=0.5, main="Optimal number of clusters",
     sub=paste("Optimal k =", k))
par(opar)
```

Index