Package: OptCirClust (via r-universe)

September 8, 2024

Type Package

Title Circular, Periodic, or Framed Data Clustering: Fast, Optimal, and Reproducible

Version 0.0.4

Date 2021-07-27

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

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

Description Fast, optimal, and reproducible clustering algorithms for circular, periodic, or framed data. The algorithms introduced here are based on a core algorithm for optimal framed clustering the authors have developed (Debnath & Song 2021) <doi:10.1109/TCBB.2021.3077573>. The runtime of these algorithms is O(K N log^2 N), where K is the number of clusters and N is the number of circular data points. On a desktop computer using a single processor core, millions of data points can be grouped into a few clusters within seconds. One can apply the algorithms to characterize events along circular DNA molecules, circular RNA molecules, and circular genomes of bacteria, chloroplast, and mitochondria. One can also cluster climate data along any given longitude or latitude. Periodic data clustering can be formulated as circular clustering. The algorithms offer a general high-performance solution to circular, periodic, or framed data clustering.

VignetteBuilder knitr

License LGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

LinkingTo Rcpp

Imports Ckmeans.1d.dp, graphics, plotrix, Rcpp, Rdpack, stats, reshape2

Suggests ape, ggplot2, knitr, rmarkdown, testthat

NeedsCompilation yes

RdMacros Rdpack

Date/Publication 2021-07-28 07:40:05 UTC

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

RemoteUrl https://github.com/cran/OptCirClust

RemoteRef HEAD

RemoteSha e176681b90869e849aa18cd54aafa265a5df7928

Contents

`CirClust` *Circular Data Clustering*

Description

Perform clustering on circular data to minimize the within-cluster sum of squared distances.

Usage

```
CirClust(O, K, Circumference, method = c("FOCC", "HEUC", "BOCC"))
```

Arguments

`O`	a vector of circular data points. They can be coordinates along the circle based on distance, or angles around the circle.
`K`	the number of clusters
`Circumference`	the circumference of the circle where data are located
`method`	the circular clustering method. `"FOCC"`: fast and optimal, the default method; `"HEUC"`: based on heuristic k-means, fast but not necessarily optimal; `"BOCC"`: brute-force based on Ckmeans.1d.dp, slow but optimal, included to provide a baseline.

Details

By circular data, we broadly refer to data points on any non-self-intersecting loop. In clustering N circular points into K clusters, the "FOCC" algorithm is reproducible with runtime $O(KN\log^2 N)$ (Debnath and Song 2021); The "HEUC" algorithm, not always reproducible, calls the `kmeans` function repeatedly; The "BOCC" algorithm with runtime $O(KN^2)$, reproducible but slow, is done via repeatedly calling the `Ckmeans.1d.dp` function.

Value

An object of class `"CirClust"` which has a `plot` method. It is a list with the following components:

`cluster`	a vector of clusters assigned to each element in `O`. Each cluster is indexed by an integer from 1 to `K`.
`centers`	a numeric vector of the means for each cluster in the circular data.
`withinss`	a numeric vector of the within-cluster sum of squares for each cluster.
`size`	a vector of the number of elements in each cluster.
`totss`	the total sum of squared distances between each element and the sample mean. This statistic is not dependent on the clustering result.
`tot.withinss`	the total sum of within-cluster squared distances between each element and its cluster mean. This statistic is minimized given the number of clusters.
`betweenss`	the sum of squared distances between each cluster mean and sample mean. This statistic is maximized given the number of clusters.
`ID`	the starting index of the frame with minimum SSQ
`Border`	the borders of `K` clusters
`Border.mid`	the middle point of the last and first points of two consequitive clusters.
`O_name`	a character string. The actual name of the `O` argument.
`Circumference`	the circumfarence of the circular or periodic data.

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

```
O <- c(1,2, 10,11,12,13,14,15, 27,28,29,30,31,32, 40,41)

K <- 3

Circumference <- 42

# Perform circular clustering:
output <- CirClust(O, K, Circumference)

# Visualize the circular clusters:
opar <- par(mar=c(1,1,2,1))
```

```
plot(output)
par(opar)
```

`FramedClust` *Framed Data Clustering*

Description

Find a frame of given size, among all possible such frames on the input data, to minimize the minimum within-cluster sum of square distances.

Usage

```
FramedClust(
  X,
  K,
  frame.size,
  first.frame = 1,
  last.frame = length(X) - frame.size + 1,
  method = c("linear.polylog", "kmeans", "Ckmeans.1d.dp")
)
```

Arguments

`X`	a vector of data points to perform framed clustering
`K`	the number of clusters in each frame
`frame.size`	the number of points from X to be included in each frame. It is not the width of the frame.
`first.frame`	starting index of the first frame to be clustered. The first point in the first frame is `X[first.frame]`.
`last.frame`	starting index of the last frame to be clustered. The first point in the first frame is `X[last.frame]`.
`method`	the framed clustering method. See Details.

Details

The method option `"linear.polylog"` (default) performs fast optimal framed clustering. The runtime is $O(KN \log^2 N)$ (Debnath and Song 2021).

The `"kmeans"` option repeatedly calling the heuristic k-means algorithm in all frames without any guarantee of cluster optimality.

The method option `"Ckmeans.1d.dp"` performs optimal framed clustering by repeatedly finding the best clustering within each frame using the `"Ckmeans.1d.dp"` method. At a runtime of $O(KN^2)$, the algorithm is slow but optimal. It is included to provide a baseline.

Value

An object of class "FramedClust" which has a plot method. It is a list with the following components:

cluster	a vector of clusters assigned to each element in x. Each cluster is indexed by an integer from 1 to K. NA represents points from X that are outside the optimal frame, thus not part of any cluster.
centers	a numeric vector of the means for each cluster in the frame.
withinss	a numeric vector of the within-cluster sum of squared distances for each cluster.
size	a vector of the number of elements in each cluster.
totss	total sum of squared distances between each element and the sample mean. This statistic is not dependent on the clustering result.
tot.withinss	total sum of within-cluster squared distances between each element and its cluster mean. This statistic is minimized given the number of clusters.
betweenss	sum of squared distances between each cluster mean and sample mean. This statistic is maximized given the number of clusters.
X_name	a character string. The actual name of the X argument.

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

```
N <- 100
X <- rnorm(N)
K <- 5
frame.size <- 60

result <- FramedClust(X, K, frame.size)
plot(result, main="Example 1. Framed clustering on all frames")

frame.size <- 40
first.frame <- 30
last.frame <- 50
method <- "linear.polylog"

result <- FramedClust(X, K, frame.size, first.frame,
                      last.frame, method)
plot(result, main="Example 2. Framed clustering on a subset of frames")
```

plot.CirClust *Plot Method for Circular Data Clustering*

Description

The plot method for circular data clustering result object of class CirClust. It visualizes circular clusters on the input data.

Usage

```
## S3 method for class 'CirClust'
plot(
  x,
  xlab = "",
  ylab = "",
  main = NULL,
  sub = "",
  col.clusters = c("blue", "red3", "green3", "orange", "purple", "brown"),
  axes = FALSE,
  xlim = c(-1.75, 1.75),
  ylim = c(-1.75, 1.75),
  fill = "floralwhite",
  border = "gray36",
  border.lty = "dotted",
  ...
)
```

Arguments

x	an object of class as returned by CirClust
xlab	a character string. The x-axis label for the plot. Default is no string.
ylab	a character string. The y-axis label for the plot. Default is no string.
main	a character string. The title for the plot.
sub	a character string. The subtitle for the plot.
col.clusters	a vector of colors, defined either by integers or by color names. If the length is shorter than the number of clusters, the colors will be reused. By default the blue, red3, green3, orange, purple, brown colors are used in the plot.
axes	the axis will be ploted if set TRUE. Default is FALSE.
xlim	range of the x axis in the plot. Default is from -1.75 to 1.75.
ylim	range of the y axis in the plot. Default is from -1.75 to 1.75.
fill	the color to fill inside the ring as the background of data points.
border	the color to draw cluster borders.
border.lty	the line type to draw cluster borders.
...	other arguments associated with the plot function

Value

A copy of the input object of class `CirClust`.

Examples

```
opar <- par(mar=c(1,1,2,1))
# Example 1. Circular data clustering
n <- 100
Circumference <- 7
O <- runif(n, 0, Circumference)
result <- CirClust(O, K=3, Circumference=Circumference)
plot(result, fill="mintcream", main="Example 1. Circular clustering")


# Example 2. Circular data clustering
n <- 40
m <- 5
O <- c(rnorm(n,mean=5,sd=m), rnorm(n,mean=15,sd=m), rnorm(n,mean=26,sd=m))
K <- 3
Circumference <- 28

result <- CirClust(O, K, Circumference, method = "FOCC")

color <- c("royalblue", "green3", "firebrick") # c("#0000CD","#808080", "#DC143C")

par(mar=c(1,1,2,1))

plot(result, col.clusters = color, fill="floralwhite",
     main="Example 2. Circular clustering")


# Example 3. Periodic data clustering
n <- 100
period <- 5.2
O <- rnorm(n)
result <- CirClust(O, K=5, Circumference=period)
plot(result, fill="navy", border="gray", border.lty="dotted",
     main="Example 3. Periodic clustering")

par(opar)
```

`plot.FramedClust` *Plot Method for Framed Data Clustering*

Description

The `plot` method for framed data clustering result object. It visualizes clusters on the input data that are within a best frame.

Usage

```
## S3 method for class 'FramedClust'
plot(
  x,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  sub = NULL,
  col.clusters = c("blue", "red3", "green3", "orange", "purple", "brown"),
  ...
)
```

Arguments

x	an object of class FramedClust as returned by FramedClust
xlab	a character string. The x-axis label for the plot. Default is NULL.
ylab	a character string. The y-axis label for the plot. Default is NULL.
main	a character string. The title for the plot. Default is NULL.
sub	a character string. The subtitle for the plot. Default is NULL.
col.clusters	a vector of colors, defined either by integers or by color names. If the length is shorter than the number of clusters, the colors will be reused. By default the blue, red3, green3, orange, purple, brown colors are used in the plot.
...	other arguments associated with the plot function

Value

An object of class "FramedClust", identical to the input x

Examples

```
N <- 100
X <- rchisq(N, 5)
K <- 3
frame.size <- 40

result <- FramedClust(X, K, frame.size)

plot(result)
```

Index