Package 'EfficientOASW'

October 25, 2023

Title Efficient Clustering Algorithms for Optimizing the Average Silhouette Width

Version 0.0.1

Author Minh Long Nguyen <edelweiss611428@gmail.com>

Maintainer Minh Long Nguyen <edelweiss611428@gmail.com>

Description This package implements the original Optimum Silhouette (OSil) clustering algorithm, an O(N) faster implementation of the exact OSil algorithm called Efficient Optimum Silhoutte (effOSil). This package also implements approximate algorithms of OSil and effOSil called Fast Optimum Silhoutte (FOSil) and Scalable Optimum Silhouette (scalOSil), respectively.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LinkingTo Rcpp

Imports Rcpp, cluster, stats

Archs x64

R topics documented:

effOSil 2
FOSil 3
Init 4
OSil 5
scalOSil 6
Silhouette 7

Index **9**

effOSil *The Efficient Optimum Silhouette algorithm*

Description

This function implements the exact Optimum Silhouette (OSil) algorithm.

Usage

```
effOSil(dx, k, initClustering = NULL, initMethod = "average")
```

Arguments

dx	A "dist" object, which can be obtained by the "dist" function.
k	The number of clusters.
initClustering	An initialized clustering. It must be an numeric vector of k unique values 1,2,...,k. By default, initClustering is set to NULL. If initClustering is NULL, initMethod is used instead; otherwise, initClustering is used.
initMethod	A character vector specifying initialization methods. It must contain only supported methods: one of the two combined methods "multiple1" and "multiple2"; or any combination of of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid". See ?Init for more details.

Details

This function implements the exact Optimum Silhouette (OSil) algorithm proposed by Batool & Hennig (2021). However, it is O(N) times faster than the original OSil algorithm at the cost of storing O(N) additional values.

Value

Clustering The OSil clustering solution.

ASW The ASW associated with the OSil clustering.

nIter The number of iterations needed for convergence.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Batool, F. and Hennig, C., 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158, p.107190.

Examples

```
x = iris[,-5]
dx = dist(x)
effOSil_clustering = effOSil(dx, 3, initMethod = "average")
plot(x, col = effOSil_clustering$Clustering)
```

FOSil *The Fast Optimum Silhouette algorithm*

Description

This function implements the Fast Optimum Silhouette (FOSil) algorithm.

Usage

```
FOSil(dx, k, n = "default", ns = 25, initMethod = "average")
```

Arguments

dx	A "dist" object, which can be obtained by the "dist" function.
k	The number of clusters.
n	The sample size. By default, n = ceiling(0.2*N).
ns	The number of scalOSil instances. By default, ns = 25.
initMethod	A character vector specifying initialization methods. It must contain only supported methods: one of the two combined methods "multiple1" and "multiple2"; or any combination of of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid". See ?Init for more details.

Details

The Fast Optimum Silhouette algorithm (FOSil; Batool & Hennig (2021)) is an approximation algorithm of OSil, based on subsetting. It consists of two steps: partial clustering (PC) and classification (C).

In the PC-step of FOSil, FOSil is applied to various subsets of equal size, in which the subset S and its OSil clustering C_S maximizing the ASW is selected. In the C-step of FOSil, each unassigned data point is classified into one of the clusters in C_S in such a way that the ASW is maximized.

However, FOSil is still a computationally expensive algorithm. The PC-step of FOSil scales cubically in n and the C-step of FOSil scales quadratically in n. scalOSil is an improved version of FOSil, which improves both steps of FOSil by O(n) time, allowing us to handle much larger datasets (see ?scalOSil for more details).

Value

Clustering Final clustering.

ASW The ASW of the scalOSil clustering w.r.t. dx.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Batool, F. and Hennig, C., 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158, p.107190.

Examples

```
x = iris[,-5]
dx = dist(x)
FOSil_clustering = FOSil(dx, 3, initMethod = "average")
plot(x, col = FOSil_clustering$Clustering)
```

`Init` *Initialization methods for Optimum Average Silhouette Width clustering algorithm*

Description

This function computes an initialized clustering for Optimum Average Silhouette Width clustering algorithms

Usage

```
Init(dx, k, initMethod = "average")
```

Arguments

`dx`	A "dist" object, which can be obtained by the "dist" function.
`k`	The number of clusters.
`initMethod`	A character vector specifying initialization methods. It must contain only supported methods: one of the two combined methods "multiple1" and "multiple2"; or any combination of of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid".

Details

This function computes an initialized clustering for Optimum Average Silhouette Width clustering algorithms by using the clustering methods specified by initMethod, and return the clustering maximize the ASW. The two combined methods "multiple1" and "multiple2" are:

"multiple1" PAM and average linkage.

"multiple2" PAM, average linkage, single linkage, and the Ward's method (ward.D2).

Value

Clustering An initialized clustering.

ASW The ASW associated with the initialized clustering.

Method The "best" initialization method.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Batool, F. and Hennig, C., 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158, p.107190. Batool, F., 2019. Initialization methods for optimum average silhouette width clustering. arXiv preprint arXiv:1910.08644.

Examples

```
x = iris[,-5]
dx = dist(x)
Init(dx,3,"multiple1")
```

`OSil` *The original Optimum Silhouette algorithm*

Description

This function implements the original Optimum Silhouette algorithm.

Usage

```
OSil(dx, k, initClustering = NULL, initMethod = "average")
```

Arguments

`dx`	A "dist" object, which can be obtained by the "dist" function.
`k`	The number of clusters.
`initClustering`	An initialized clustering. It must be an numeric vector of k unique values 1,2,...,k. By default, initClustering is set to NULL. If initClustering is NULL, initMethod is used instead; otherwise, initClustering is used.
`initMethod`	A character vector specifying initialization methods. It must contain only supported methods: one of the two combined methods "multiple1" and "multiple2"; or any combination of of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid". See ?Init for more details.

Details

This function implements the original, computationally expensive Optimum Silhouette algorithm (Batool & Hennig 2021).

Value

Clustering The OSil clustering solution.

ASW The ASW associated with the OSil clustering.

nIter The number of iterations needed for convergence.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Batool, F. and Hennig, C., 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158, p.107190.

Examples

```
x = iris[,-5]
dx = dist(x)
OSil_clustering = OSil(dx, 3, initMethod = "average")
plot(x, col = OSil_clustering$Clustering)
```

`scalOSil` *The Scalable Optimum Silhouette algorithm*

Description

This function implements the Scalable Optimum Silhouette (scalOSil) algorithm.

Usage

```
scalOSil(dx, k, n = "default", rep = 5, initMethod = "average")
```

Arguments

`dx`	A "dist" object, which can be obtained by the "dist" function.
`k`	The number of clusters.
`n`	The sample size. By default, n = ceiling(0.2*N).
`rep`	The number of scalOSil instances. By default, rep = 5.
`initMethod`	A character vector specifying initialization methods. It must contain only supported methods: one of the two combined methods "multiple1" and "multiple2"; or any combination of of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid". See ?Init for more details.

Details

The scalOSil algorithm is an approximation algorithm of effOSil based on subsetting. It is an improved version of FOSil (Batool & Hennig 2021). Both the algorithms consists of two steps: partial clustering (PC) and classification (C).

In the PC-step of scalOSil, effOSil is applied to a random subset of the dataset, obtaining a subset S and its effOSil clustering. In the C-step of scalOSil, each unassigned data point is classified into one of the clusters in C_S in such a way that the ASW is maximized.

Unlike FOSil, scalOSil runs many instances, specified by the parameter "rep", and for each instance, scalOSil only runs the PC-step once. Moreover, the PC-step of scalOSil scales quadratically in n, while that of FOSil scales cubically in n, and the C-step of scalOSil scales linearly in n, while that of FOSil scales quadratically in n. These allow scalOSil to handle much larger datasets.

Value

Clustering Final clustering.

ASW The ASW of the scalOSil clustering w.r.t. dx.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Batool, F. and Hennig, C., 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158, p.107190.

Examples

```
x = iris[,-5]
dx = dist(x)
scalOSil_clustering = scalOSil(dx, 3, initMethod = "average")
plot(x, col = scalOSil_clustering$Clustering)
```

`Silhouette` *Silhouette Width computation*

Description

This function computes the Silhouette Widths of all data points in the dataset.

Usage

```
Silhouette(C, dx)
```

Arguments

`C`	A clustering solution. It must be an integer vector of k unique values 1,2,...,k.
`dx`	A "dist" object, which can be obtained by the "dist" function.

Value

A numeric matrix of class "silhouette" containing three columns

cluster A clustering of the dataset.

neighbor The clustering labels of the nearest clusters for all data points.

sil_width The silhouette widths of data points.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Rousseeuw, P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math., 20, 53–65.

Examples

```
library("cluster")
x = iris[,-5]
dx = dist(x)
C = pam(dx, 3)$clustering
Silhouette(C,dx)
```

Index

effOSil, 2

FOSil, 3

Init, 4

OSil, 5

scalOSil, 6
Silhouette, 7