Package ‘ASW’

October 30, 2024

Title Clustering Algorithms for Optimising the Average Silhouette Width

Version 0.0.2

Author Minh Long Nguyen <edelweiss611428@gmail.com>

Maintainer Minh Long Nguyen <edelweiss611428@gmail.com>

Description
This package implements clustering algorithms for optimising the Average Silhouette Width.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo Rcpp

Imports Rcpp,
cluster,
stats

Contents

`effOSil` *The Efficient Optimum Silhouette algorithm*

Description

This function implements the Efficient Optimum Silhouette (effOSil) algorithm.

Usage

```
effOSil(dx, K, initMethod, variant)
```

Arguments

`dx`	A dist object, which can be computed using the stats::dist() function.
`K`	An integer vector specifying the number of clusters. By default, K = 2:12.
`initMethod`	A character vector specifying initialisation methods. By default, initMethod = "average"; however, to achieve the best initialisation in terms of the ASW, various initialisation methods should be used (e.g., initMethod = c("single", "average", "complete", "pam")). See ?Init for more details.
`variant`	An algorithmic variant. Options include "efficient" and "original". By default, variant = "efficient", indicating that effOSil is used. If variant = "original", the original, computationally expensive OSil algorithm is used.

Details

This function implements the Efficient Optimum Silhouette (effOSil) algorithm, an O(N) runtime improvement of the original, computationally expensive Fast OSil (FOSil) algorithm proposed by Batool & Hennig (2021) where N is the number of observations. This function also implements the OSil algorithm for comparision purporses.

Value

best_clustering The clustering achieving the highest ASW value.

best_asw The highest ASW value.

k The estimated number of clusters.

clusterings The effOSil clusterings for all k in K.

asw The ASW values associated with the clusterings.

nIter The numbers 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 = scale(faithful)
dx = dist(x)
effOSil_clustering = effOSil(dx = dx, K = 2:12)
par(mfrow = c(2,1))
plot(faithful, col = effOSil_clustering$best_clustering, pch = 4)
plot(2:12, effOSil_clustering$asw, type = "l", xlab = "k", ylab = "ASW")
par(mfrow = c(1,1))
```

`Init` *Initialisation methods for the Optimum Silhouette algorithm.*

Description

This function computes an initialisation for the Optimum Silhouette algorithm.

Usage

```
Init(dx, k, initMethod)
```

Arguments

`dx` dx A dist object, which can be computed using the stats::dist() function.

`k` An integer specifying the number of clusters.

`initMethod` A character vector (or string) specifying initialisation methods. Options include any combination of "pam", "average", "single", "complete", "ward.D", "ward.D2", "mcquitty", "median", and "centroid". By default, initMethod = "average".

Details

This function computes an initialisation for the Optimum Silhouette algorithm, but it can be used as a stand-alone clustering method (i.e., run different clustering algorithms and select the clustering solution maximising the ASW).

Value

clustering An initialised clustering.

asw The ASW associated with the initialised clustering.

method The "best" initialisation 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 = faithful
dx = dist(x)
InitClustering = Init(dx, 2, c("pam", "average", "complete"))
plot(x, col = InitClustering$clustering, pch = 4)
print(paste(InitClustering$method, "achieves the highest ASW value"))
```

PAMSil *The PAMSil algorithm*

Description

This function implements the PAMSil algorithm.

Usage

```
PAMSil(dx, K)
```

Arguments

dx	A dist object, which can be computed using the stats::dist() function.
K	An integer vector specifying the number of clusters. By default, K = 2:12.

Details

This function implements the PAMSil algorithm proposed by Van der Laan et al. (2003), a k-medoid clustering algorithm whose objective function is the ASW.

Value

best_clustering The clustering achieving the highest ASW value.

best_asw The highest ASW value.

best_medoids The medoids associated with the clustering maximizing the ASW.

k The estimated number of clusters.

clusterings The PAMSil clusterings for all k in K.

asw The ASW values associated with the clusterings.

medoids The medoids associated with the clustering solutions.

nIter The numbers of iterations needed for convergence.

Author(s)

Minh Long Nguyen <edelweiss611428@gmail.com>

References

Van der Laan, M., Pollard, K. and Bryan, J., 2003. A new partitioning around medoids algorithm. Journal of Statistical Computation and Simulation, 73(8), pp.575-584.

Examples

```
x = scale(faithful)
dx = dist(x)
PAMSil_clustering = PAMSil(dx = dx, K = 2:12)
par(mfrow = c(2,1))
plot(faithful, col = PAMSil_clustering$best_clustering, pch = 4)
plot(2:12, PAMSil_clustering$asw, type = "l", xlab = "k", ylab = "ASW")
par(mfrow = c(1,1))
```

scalOSil *The Scalable Optimum Silhouette Algorithm*

Description

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

Usage

```
scalOSil(dx, K, n, ns, rep, initMethod, variant)
```

Arguments

`dx` A dist object, which can be computed using the stats::dist() function.

`K` An integer vector specifying the number of clusters. By default, K = 2:12.

`n` An integer specifying the sample size. If not specified, n = 0.1*N where N is the number of observations.

`ns` An integer specifying the number of random samples in each instance. By default, ns = 10.

`rep` An integer specifying the number of scalOSil instances. By default, rep = 1.

`initMethod` A character vector specifying initialisation methods. By default, initMethod = "average"; however, to achieve the best initialisation in terms of the ASW, various initialisation methods should be used (e.g., initMethod = c("single", "average", "complete", "pam")). See ?Init for more details.

`variant` An algorithmic variant. Options include "scalable" and "original". By default, variant = "scalable", indicating that scalOSil is used. If variant = "original", the original, computationally expensive FOSil algorithm is used.

Details

This function implements the Scalable Optimum Silhouette (scalOSil) algorithm, an O(n) runtime improvement of the original, computationally expensive Fast OSil (FOSil) algorithm proposed by Batool & Hennig (2021) where n is the sub-sample size. This function also implements the FOSil algorithm for comparision purporses.

Value

best_clustering The clustering achieving the highest ASW value.

best_asw The highest ASW value.

k The estimated number of clusters.

clusterings The scalOSil clusterings for all k in K.

asw The ASW values associated with the clusterings.

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 = scale(faithful)
dx = dist(x)
scalOSil_clustering = scalOSil(dx = dx, K = 2:12, n = ceiling(0.25*nrow(x)), ns = 10)
set.seed(59)
par(mfrow = c(2,1))
plot(faithful, col = scalOSil_clustering$best_clustering, pch = 4)
plot(2:12, scalOSil_clustering$asw, type = "l", xlab = "k", ylab = "ASW")
par(mfrow = c(1,1))
```

`Silhouette` *Silhouette Width*

Description

This function computes the Silhouette Widths given a distance matrix and a clustering solution.

Usage

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

Arguments

`C`	An integer vector specifying a clustering solution. min(C) must be 1 and max(C) must be k.
`dx`	A dist object, which can be computed using the stats::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")
dx = dist(faithful)
C = pam(dx, 2)$clustering
plot(Silhouette(C,dx))
```

Index