Package 'fastkmedoids'

October 13, 2022

Type Package

Title Faster K-Medoids Clustering Algorithms: FastPAM, FastCLARA, FastCLARANS

Version 1.2

Date 2021-01-13

Author Xun Li

Maintainer Xun Li <lixun910@gmail.com>

Description R wrappers of C++ implementation of Faster K-Medoids clustering algorithms (FastPAM, FastCLARA and FastCLARANS) proposed in Erich Schubert, Peter J. Rousseeuw 2019 <doi:10.1007/978-3-030-32047-8_16>.

Depends methods

License GPL (>= 2)

Collate RcppExports.R KmedoidsResult.R

Encoding UTF-8

Imports Rcpp (>= 1.0.1)

LinkingTo Rcpp

RoxygenNote 7.1.1

SystemRequirements C++11

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-01-21 17:20:06 UTC

R topics documented:

`fastkmedoids-package` *Faster K-Medoids Clustering Algorithms: FastPAM, FastCLARA, FastCLARANS*

Description

This package provides R wrappers of C++ implementation of Faster K-Medoids clustering algorithms (FastPAM, FastCLARA and FastCLARANS) proposed in Erich Schubert and Peter J. Rousseeuw 2019.

Details

The C++ Faster K-Medoids clustering algorithms (FastPAM, FastCLARA and FastCLARANS) are ported from ELKI project (see http://elki-project.github.io/). To generate identical results, the random number generator, specifically the xorshift+ generator, is also ported. The results between this fastkmedoids R package should be the same with ELKI if using same initial seed for random number generator.

Besides FastPAM, FastCLARA and FastCLARANS, the classic algorithms, including PAM, CLARA and CLARANS, are also implemented. If interested in writing wrappers for these algorithms, please use the github repository: https://github.com/lixun910/fastkmedoids

All three algorithms take the distance matrix (lower triangular part, column wise storage) as input, which can be computed using dist() function in R (see the examples below). If using a pre-computed distance matrix, please transform it (lower triangular part, column wise storage) to a 1-dimensional array.

All three algorithms takes the same parameters as in ELKI. If the explanation of the input paramters is not clear, please refer to ELKI :

FastPAM: https://elki-project.github.io/releases/current/javadoc/de/lmu/ifi/dbs/elki/algorithm/clustering/kmeans/KMedoidsF
FastCLARA: https://elki-project.github.io/releases/current/javadoc/de/lmu/ifi/dbs/elki/algorithm/clustering/kmeans/FastCLA
FastCLARANS: https://elki-project.github.io/releases/current/javadoc/de/lmu/ifi/dbs/elki/algorithm/clustering/kmeans/FastC

The C++ code is a part of GeoDa (https://github.com/geodacenter/geoda) and libgeoda. If you are interested in a GUI version of this C++ implementation. You can download and use the free and cross-platform GeoDa software from https://geodacenter.github.io. The lab note of using K-Medoids in GeoDa is here: https://geodacenter.github.io/workbook/7c_clusters_3/lab7c.html#k-medoids.

Author(s)

Xun Li Maintainer: Xun Li <lixun910@gmail.com>

References

Erich Schubert, Peter J. Rousseeuw "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms" 2019 <doi:10.1007/978-3-030-32047-8_16>

See Also

https://arxiv.org/abs/1810.05691

Examples

```
# We use the demo data sets "USArrests"
data("USArrests")
df <- scale(USArrests)
dv <- as.vector(dist(df)) # compute distance matrix (lower triangular part)
n <- nrow(df)

# PAM
clusters <- pam(dv, n, k=3)
clusters

# FastPAM (use "LAB" initializer by default)
clusters1 <- fastpam(dv, n, k=3)
clusters1

# FastPAM, specify "BUILD" as initializer
#clusters2 <- fastpam(dv, n, k=3, initializer="BUILD")
#clusters2

# FastCLARA
#clusters3 <- fastclara(dv, n, k=3, numsamples = 5, sampling=0.25)
#clusters3

# FastCLARANS
#clusters4 <- fastclarans(dv, n, k=3, numlocal=2, maxneighbor=0.025)
#clusters4
```

`fastclara` *FastCLARA*

Description

Clustering Large Applications (CLARA) with the improvements, to increase scalability in the number of clusters. This variant will also default to twice the sample size, to improve quality. (Schubert and Rousseeuw, 2019)

Usage

```
fastclara(
  rdist,
  n,
  k,
  maxiter = 0L,
  initializer = "LAB",
  fasttol = 1,
  numsamples = 5L,
  sampling = 0.25,
  independent = FALSE,
  seed = 123456789L
)
```

Arguments

rdist	The distance matrix (lower triangular matrix, column wise storage)
n	The number of observations
k	The number of clusters to produce
maxiter	The maximum number of iterations (default: 0)
initializer	Initializer: either "BUILD" (used in classic PAM) or "LAB" (linear approximative BUILD)
fasttol	Tolerance for fast swapping behavior (may perform worse swaps). Default: 1.0, which means to perform any additional swap that gives an improvement. When set to 0, it will only execute an additional swap if it appears to be independent (i.e., the improvements resulting from the swap have not decreased when the first swap was executed).
numsamples	Number of samples to draw (i.e. iterations). Default: 5
sampling	Sampling rate. Default value: 80 + 4*k. (see Schubert and Rousseeuw, 2019) If less than 1, it is considered to be a relative value. e.g. N*0.10
independent	NOT Keep the previous medoids in the next sample. Default: FALSE
seed	Seed for random number generator. Default: 123456789

Value

KMedoids S4 class

References

Erich Schubert, Peter J. Rousseeuw "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms" 2019 https://arxiv.org/abs/1810.05691

---

fastclarans *FastCLARANS*

Description

A faster variation of CLARANS, that can explore O(k) as many swaps at a similar cost by considering all medoids for each candidate non-medoid. Since this means sampling fewer non-medoids, we suggest to increase the subsampling rate slightly to get higher quality than CLARANS, at better runtime. (Schubert and Rousseeuw, 2019)

Usage

```
fastclarans(rdist, n, k, numlocal = 2L, maxneighbor = 0.025, seed = 123456789L)
```

Arguments

rdist	The distance matrix (lower triangular matrix, column wise storage)
n	The number of observations
k	The number of clusters to produce.
numlocal	Number of samples to draw (i.e. restarts). Default: 2
maxneighbor	Sampling rate. If less than 1, it is considered to be a relative value. Default: 2 * 0.0125, larger sampling rate than CLARANS (see Schubert and Rousseeuw, 2019)
seed	Seed for random number generator. Default: 123456789

Value

KMedoids S4 class

References

Erich Schubert, Peter J. Rousseeuw "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms" 2019 https://arxiv.org/abs/1810.05691

---

fastpam *FastPAM*

Description

FastPAM: An improved version of PAM, that is usually O(k) times faster. Because of the speed benefits, we also suggest to use a linear-time initialization, such as the k-means++ initialization or the proposed LAB (linear approximative BUILD, the third component of FastPAM) initialization, and try multiple times if the runtime permits. (Schubert and Rousseeuw, 2019)

Usage

```
fastpam(
  rdist,
  n,
  k,
  maxiter = 0L,
  initializer = "LAB",
  fasttol = 1,
  seed = 123456789L
)
```

Arguments

rdist	The distance matrix (lower triangular matrix, column wise storage)
n	The number of observations
k	The number of clusters to produce.
maxiter	The maximum number of iterations (default: 0)
initializer	Initializer: either "BUILD" (used in classic PAM) or "LAB" (linear approximative BUILD) Because of the speed benefits, "LAB" is suggested, and one can try multiple times if the runtime permits.
fasttol	Tolerance for fast swapping behavior (may perform worse swaps). Default: 1.0, which means to perform any additional swap that gives an improvement. When set to 0, it will only execute an additional swap if it appears to be independent (i.e., the improvements resulting from the swap have not decreased when the first swap was executed).
seed	Seed for random number generator. Default: 123456789

Value

KMedoids S4 class

References

Erich Schubert, Peter J. Rousseeuw "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms" 2019 https://arxiv.org/abs/1810.05691

KmedoidsResult-class *An S4 class to represent the result of kmedoids clustering*

Description

An S4 class to represent the result of kmedoids clustering

Slots

`cost` The cost value of kmedoids clustering

`medoids` The medoids of kmedoids clustering

`assignment` The assignment of which cluster each observation belongs to of kmedoids clustering

pam *PAM (Partitioning Around Medoids)*

Description

The original Partitioning Around Medoids (PAM) algorithm or k-medoids clustering, as proposed by Kaufman and Rousseeuw; a largely equivalent method was also proposed by Whitaker in the operations research domain, and is well known by the name "fast interchange" there. (Schubert and Rousseeuw, 2019)

Usage

```
pam(rdist, n, k, maxiter = 0L)
```

Arguments

rdist	The distance matrix (lower triangular matrix, column wise storage)
n	The number of observations
k	The number of clusters to produce
maxiter	The maximum number of iterations (default: 0)

Value

KMedoids S4 class

References

L. Kaufman, P. J. Rousseeuw "Clustering by means of Medoids" Information Systems and Operational Research 21(2)

Index