Package 'clusternor'

March 29, 2019

Version 0.0-3

Date 2019-03-28

Title A Parallel Clustering Non-Uniform Memory Access ('NUMA') Optimized Package

Description The clustering 'NUMA' Optimized Routines package or 'clusternor' is a highly optimized package for performing clustering in parallel with accelerations specifically targeting multi-core Non-Uniform Memory Access ('NUMA') hardware architectures. Disa Mhembere, Da Zheng, Carey E. Priebe, Joshua T. Vogelstein, Randal Burns (2019) <arXiv:1902.09527>.

LinkingTo Rcpp

Depends R (>= 3.0), Rcpp (>= 0.12.8)

License Apache License 2.0

URL https://github.com/neurodata/knorR

SystemRequirements GNU make C++11, pthreads

BugReports https://github.com/flashxio/knor/issues

RoxygenNote 6.1.1

Encoding UTF-8

LazyData true

NeedsCompilation yes

Suggests testthat

Author Disa Mhembere [aut, cre], Neurodata (https://neurodata.io) [cph]

Maintainer Disa Mhembere <disa@cs.jhu.edu>

Repository CRAN

Date/Publication 2019-03-29 15:40:03 UTC

R topics documented:

`FuzzyCMeans` *Perform Fuzzy C-means clustering on a data matrix. A soft variant of the kmeans algorithm where each data point are assigned a contribution weight to each cluster*

Description

See: https://en.wikipedia.org/wiki/Fuzzy_clustering#Fuzzy_C-means_clustering

Usage

```
FuzzyCMeans(data, centers, nrow = -1, ncol = -1,
  iter.max = .Machine$integer.max, nthread = -1, fuzz.index = 2,
  init = c("forgy", "none"), tolerance = 1e-06,
  dist.type = c("sqeucl", "eucl", "cos", "taxi"))
```

Arguments

`data`	Data file name on disk (NUMA optimized) or In memory data matrix
`centers`	Either (i) The number of centers (i.e., k), or (ii) an In-memory data matrix
`nrow`	The number of samples in the dataset
`ncol`	The number of features in the dataset
`iter.max`	The maximum number of iteration of k-means to perform
`nthread`	The number of parallel threads to run
`fuzz.index`	The fuzziness coefficient/index (> 1 and < inf)
`init`	The type of initialization to use c("forgy", "none")
`tolerance`	The convergence tolerance
`dist.type`	What dissimilarity metric to use

Value

A list containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations. contrib.mat: The data point to cluster contribution matrix

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
k <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
fcm <- FuzzyCMeans(iris.mat, k, iter.max=5)
```

Hmeans *Perform parallel hierarchical clustering on a data matrix.*

Description

A recursive (not acutally implemented as recursion) partitioning of data into two disjoint sets at every level as described in https://en.wikipedia.org/wiki/Hierarchical_clustering

Usage

```
Hmeans(data, kmax, nrow = -1, ncol = -1, iter.max = 20,
  nthread = -1, init = c("forgy"), tolerance = 1e-06,
  dist.type = c("eucl", "cos", "sqeucl", "taxi"), min.clust.size = 1)
```

Arguments

`data`	Data file name on disk (NUMA optmized) or In memory data matrix
`kmax`	The maximum number of centers
`nrow`	The number of samples in the dataset
`ncol`	The number of features in the dataset
`iter.max`	The maximum number of iteration of k-means to perform
`nthread`	The number of parallel threads to run
`init`	The type of initialization to use c("forgy") or initial centers
`tolerance`	The convergence tolerance for k-means at each hierarchical split
`dist.type`	What dissimilarity metric to use
`min.clust.size`	The minimum size of a cluster when it cannot be split

Value

A list of lists containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
kmax <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
kms <- Hmeans(iris.mat, kmax)
```

Kmeans *Perform k-means clustering on a data matrix.*

Description

K-means provides **k** disjoint sets for a dataset using a parallel and fast NUMA optimized version of Lloyd's algorithm. The details of which are found in this paper https://arxiv.org/pdf/1606.08905.pdf.

Usage

```
Kmeans(data, centers, nrow = -1, ncol = -1,
  iter.max = .Machine$integer.max, nthread = -1, init = c("kmeanspp",
  "random", "forgy", "none"), tolerance = 1e-06, dist.type = c("eucl",
  "sqeucl", "cos", "taxi"))
```

Arguments

data	Data file name on disk (NUMA optimized) or In memory data matrix
centers	Either (i) The number of centers (i.e., k), or
nrow	The number of samples in the dataset
ncol	The number of features in the dataset
iter.max	The maximum number of iteration of k-means to perform
nthread	The number of parallel threads to run (ii) an In-memory data matrix, or (iii) A 2-Element *list* with element 1 being a filename for precomputed centers, and element 2 the number of centroids.
init	The type of initialization to use c("kmeanspp", "random", "forgy", "none")
tolerance	The convergence tolerance
dist.type	What dissimilarity metric to use

Value

A list containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
k <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
kms <- Kmeans(iris.mat, k)
```

KmeansPP *Perform the k-means++ clustering algorithm on a data matrix.*

Description

A parallel and scalable implementation of the algorithm described in Ostrovsky, Rafail, et al. "The effectiveness of Lloyd-type methods for the k-means problem." Journal of the ACM (JACM) 59.6 (2012): 28.

Usage

```
KmeansPP(data, centers, nrow = -1, ncol = -1, nstart = 1,
  nthread = -1, dist.type = c("sqeucl", "eucl", "cos", "taxi"))
```

Arguments

data	Data file name on disk (NUMA optimized) or In memory data matrix
centers	The number of centers (i.e., k)
nrow	The number of samples in the dataset
ncol	The number of features in the dataset
nstart	The number of iterations of kmeans++ to run
nthread	The number of parallel threads to run
dist.type	What dissimilarity metric to use c("taxi", "eucl", "cos")

Value

A list containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. energy: The sum of distances for each sample from it's closest cluster. best.start: The sum of distances for each sample from it's closest cluster.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
k <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
nstart <- 3
km <- KmeansPP(iris.mat, k, nstart=nstart)
```

`MiniBatchKmeans` *A randomized dataset sub-sample algorithm that approximates the k-means algorithm. See: https://www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf*

Description

A randomized dataset sub-sample algorithm that approximates the k-means algorithm. See: https://www.eecs.tufts.edu/~dscu

Usage

```
MiniBatchKmeans(data, centers, nrow = -1, ncol = -1,
  batch.size = 100, iter.max = .Machine$integer.max, nthread = -1,
  init = c("kmeanspp", "random", "forgy", "none"), tolerance = 0.01,
  dist.type = c("sqeucl", "eucl", "cos", "taxi"),
  max.no.improvement = 3)
```

Arguments

`data`	Data file name on disk (NUMA optimized) or In memory data matrix
`centers`	Either (i) The number of centers (i.e., k), or (ii) an In-memory data matrix, or (iii) A 2-Element *list* with element 1 being a filename for precomputed centers, and element 2 the number of centroids.
`nrow`	The number of samples in the dataset
`ncol`	The number of features in the dataset
`batch.size`	Size of the mini batches
`iter.max`	The maximum number of iteration of k-means to perform
`nthread`	The number of parallel threads to run
`init`	The type of initialization to use c("kmeanspp", "random", "forgy", "none")
`tolerance`	The convergence tolerance
`dist.type`	What dissimilarity metric to use
`max.no.improvement`	Control early stopping based on the consecutive number of mini batches that does not yield an improvement on the smoothed inertia

Value

A list containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
k <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
kms <- MiniBatchKmeans(iris.mat, k, batch.size=5)
```

`Skmeans` *Perform spherical k-means clustering on a data matrix. Similar to the k-means algorithm differing only in that data features are min-max normalized the dissimilarity metric is Cosine distance.*

Description

Perform spherical k-means clustering on a data matrix. Similar to the k-means algorithm differing only in that data features are min-max normalized the dissimilarity metric is Cosine distance.

Usage

```
Skmeans(data, centers, nrow = -1, ncol = -1,
  iter.max = .Machine$integer.max, nthread = -1, init = c("kmeanspp",
  "random", "forgy", "none"), tolerance = 1e-06)
```

Arguments

`data`	Data file name on disk (NUMA optmized) or In-memory data matrix
`centers`	Either (i) The number of centers (i.e., k), or (ii) an In-memory data matrix
`nrow`	The number of samples in the dataset
`ncol`	The number of features in the dataset
`iter.max`	The maximum number of iteration of k-means to perform
`nthread`	The number of parallel threads to run
`init`	The type of initialization to use c("kmeanspp", "random", "forgy", "none")
`tolerance`	The convergence tolerance

Value

A list containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
k <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
km <- Skmeans(iris.mat, k)
```

`test_centroids` *A small example of centroids of dim: (8,5) used as for micro-benchmarks of the clusternor package. The data are randomly generated.*

Description

A small example of centroids of dim: (8,5) used as for micro-benchmarks of the clusternor package. The data are randomly generated.

Usage

```
data(test_centroids)
```

Format

An object of class `"matrix"`

Examples

```
data(test_centroids)
kms <- Kmeans(test_data, test_centroids)
```

test_data *A small dataset of dim: (50,5) used as for micro-benchmarks of the clusternor package. The data are randomly generated hence a clear number of clusters will be hard to find.*

Description

A small dataset of dim: (50,5) used as for micro-benchmarks of the clusternor package. The data are randomly generated hence a clear number of clusters will be hard to find.

Usage

```
data(test_data)
```

Format

An object of class "matrix"

Examples

```
ncenters <- 8
kms <- Kmeans(test_data, ncenters)
```

Xmeans *Perform a parallel hierarchical clustering using the x-means algorithm*

Description

A recursive (not acutally implemented as recursion) partitioning of data into two disjoint sets at every level as described in: http://cs.uef.fi/~zhao/Courses/Clustering2012/Xmeans.pdf

Usage

```
Xmeans(data, kmax, nrow = -1, ncol = -1, iter.max = 20,
  nthread = -1, init = c("forgy"), tolerance = 1e-06,
  dist.type = c("eucl", "cos", "taxi"), min.clust.size = 1)
```

Arguments

data	Data file name on disk (NUMA optmized) or In memory data matrix
kmax	The maximum number of centers
nrow	The number of samples in the dataset
ncol	The number of features in the dataset
iter.max	The maximum number of iteration of k-means to perform

`nthread`	The number of parallel threads to run
`init`	The type of initialization to use c("forgy") or initial centers
`tolerance`	The convergence tolerance for k-means at each hierarchical split
`dist.type`	What dissimilarity metric to use
`min.clust.size`	The minimum size of a cluster when it cannot be split

Value

A list of lists containing the attributes of the output. cluster: A vector of integers (from 1:**k**) indicating the cluster to which each point is allocated. centers: A matrix of cluster centres. size: The number of points in each cluster. iter: The number of (outer) iterations.

Author(s)

Disa Mhembere <disa@cs.jhu.edu>

Examples

```
iris.mat <- as.matrix(iris[,1:4])
kmax <- length(unique(iris[, dim(iris)[2]])) # Number of unique classes
xms <- Xmeans(iris.mat, kmax)
```

Index