Package 'knor'

September 13, 2018

Version 0.0-6

Date 2018-10-09

Title Non-Uniform Memory Access ('NUMA') Optimized, Parallel K-Means

Description The k-means 'NUMA' Optimized Routine library or 'knor' is a highly optimized and fast library for computing k-means in parallel with accelerations for Non-Uniform Memory Access ('NUMA') architectures.

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 6.0.1

Encoding UTF-8

LazyData true

NeedsCompilation yes

Suggests testthat

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

Maintainer Disa Mhembere <disa@jhu.edu>

Repository CRAN

Date/Publication 2018-09-13 05:00:02 UTC

R topics documented:

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",
  "cos"), omp = FALSE, numa.opt = FALSE)
```

Arguments

data	Data file name on disk 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
iter.max	The maximum number of iteration of k-means to perform
nthread	The number of parallel thread 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
omp	Use (slower) OpenMP threads rather than pthreads
numa.opt	When passing *data* as an in-memory data matrix you can optimize memory placement for Linux NUMA machines. **NOTE:** performance may degrade with very large data & it requires 2*memory of that without this.

Value

A list containing the attributes of the output of kmeans. 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@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)
```

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

Description

A small example of centroids of dim: (8,5) used as for micro-benchmarks of the knor 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 knor 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 knor 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)
```

Index