

Package ‘kmeans.ddR’

August 29, 2016

Type Package

Title Distributed k-Means for Big Data using 'ddR' API

Version 0.1.0

Date 2015-10-28

Author Vishrut Gupta, Arash Fard

Maintainer Vishrut Gupta <vishrut.gupta@hpe.com>

Depends R (>= 3.0.0)

Imports ddR, Rcpp (>= 0.11.0)

Suggests testthat

LinkingTo Rcpp

Description Distributed k-means clustering algorithm written using 'ddR' (Distributed Data Structures) API in the 'ddR' package.

License GPL (>= 2) | file LICENSE

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-11-05 08:31:48

R topics documented:

kmeans.ddR-package	2
dkmeans	2
predict.dkmeans	4

Index	6
--------------	----------

kmeans.ddR-package *Distributed clustering for Big Data*

Description

kmeans.ddR provides a distributed training and predict functions for kmeans

Details

Package: kmeans.ddR
 Type: Package
 Version: 0.1.0
 Date: 2015-10-28

Main Functions:

- dkmeans: It is a distributed version of kmeans.

Author(s)

Vishrut Gupta, Arash Fard

dkmeans *Distributed kmeans*

Description

dkmeans function is intended to be a distributed alternative for kmeans function.

Usage

```
dkmeans(X, centers, iter.max = 10, nstart = 1,
        sampling_threshold = 1e+06, trace = FALSE,
        na_action = c("exclude", "fail"),
        completeModel = FALSE)
```

Arguments

<code>X</code>	a darray (dense or sparse) which contains the samples.
<code>centers</code>	either the number of clusters, say k , or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) samples in X is chosen as the initial centres.
<code>iter.max</code>	the maximum number of iterations allowed.

nstart	when the value specified for 'centers' argument is a number, clustering will be performed several times and the best result is reported. The best result would be the one with highest value of 'withins' regardless of its number of iterations. 'nstart' gives the number of times that a random set of centers is chosen and clustering is performed. When 'centers' argument is a matrix of centers, or when completeModel=TRUE 'nstart' will be discarded.
sampling_threshold	threshold for the method which Randomly finds centers (centralized or distributed). It should be always smaller than 1e9. When (blockSize > sampling_threshold nSample > 1e9), the distributed sampling is selected, in which first a set of blocks are randomly chosen, and then the centers are randomly selected from the samples of those blocks. Here, blockSize is the number of samples in each partition of X, and nSample is the total number of samples in X.
trace	when this argument is true, intermediate steps of the progress are displayed.
na_action	it indicates what should happen when the data contain missed values. Values of NA, NaN, and Inf in samples are treated as missed values. There are two options for this argument exclude and fail. When exclude is selected (the default choice), any sample with missed values will be ignored in the clustering process. In the darray which will be created for cluster, the value corresponding to these samples will be NA. When fail is selected, the function will stop in the case of any missed value in the dataset.
completeModel	when it is FALSE (default), the function does not return cluster label of the samples and measurements of cluster quality. Therefore, it can perform faster.

Details

The data given by X is clustered by the k-means method, which aims to partition the points into k groups such that the sum of squares from points to the assigned cluster centres is minimized. At the minimum, all cluster centres are at the mean of their Voronoi sets (the set of data points which are nearest to the cluster centre).

The algorithm of Lloyd-Forgy (Lloyd 1957 and Forgy 1965) is used at the current version. If an initial matrix of centres is supplied, it is possible that no point will be closest to one or more centres, which currently generates a warning message.

Value

dkmeans returns an object of class "dkmeans" which has a print and a fitted method. It is a list with components:

cluster	(available only when completeModel=TRUE; otherwise it is NULL) a darray of integers (from 1:k) indicating the cluster to which each point is allocated.
centers	a matrix of cluster centres.
totss	(available only when completeModel=TRUE; otherwise it is NA) the total sum of squares.
withins	(available only when completeModel=TRUE; otherwise it is NA) vector of within-cluster sum of squares, one component per cluster.

tot.withinss	(available only when completeModel=TRUE; otherwise it is NA) total within-cluster sum of squares, i.e., sum(withinss).
betweenss	(available only when completeModel=TRUE; otherwise it is NA) the between-cluster sum of squares, i.e. totss-tot.withinss.
size	the number of points in each cluster.
iter	the number of iterations used for clustering. Its value will be iter.max+1 when the algorithm is not converged.

Author(s)

Vishrut Gupta, Arash Fard

References

- Forgy, E. W. (1965) Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics* 21, 768-769.
- Lloyd, S. P. (1957, 1982) Least squares quantization in PCM. Technical Note, Bell Laboratories. Published in 1982 in *IEEE Transactions on Information Theory* 28, 128-137.

See Also

[kmeans](#)

Examples

```
## Not run:
library(kmeans.ddR)

iris2 <- iris[,-5]
centers <- matrix(c(5.901613,5.006000,6.850000,2.748387,3.428000,
3.073684,4.393548,1.462000,5.742105,1.433871,0.246000,2.071053),3,4)
dimnames(centers) <- list(1L:3L, colnames(iris2))

X <- as.darray(data.matrix(iris2))

(mykm1 <- dkmeans(X,centers=centers))

(mykm2 <- dkmeans(X,centers=3, completeModel=TRUE))

## End(Not run)
```

predict.dkmeans

distributed predict method for a kmeans model

Description

This function finds cluster label of a set of samples according to a given set of centers specified in the model object.

Usage

```
## S3 method for class 'dkmeans'  
predict(object, newdata, trace=FALSE, ...)
```

Arguments

<code>object</code>	an object returned by <code>dkmeans</code> function
<code>newdata</code>	a darray (dense or sparse) or a matrix which contains the samples.
<code>trace</code>	when this argument is true, intermediate steps of the progress are displayed.
<code>...</code>	additional arguments to be passed (None are supported).

Details

This function applies the centers found by `dkmeans` on a new set of samples in order to label them.

Value

This function returns a darray or a matrix based on the type of `newdata` which contains the corresponding label of each sample.

Author(s)

Vishrut Gupta, Arash Fard

Examples

```
## Not run:  
iris2 <- iris  
iris2$Species <- NULL  
  
library(kmeans.ddR)  
  
X <- as.darray(data.matrix(iris2))  
  
mykm <- dkmeans(X,centers=3)  
  
newdata <- matrix(c(5,4,3,5,7,1,0,8),2,4)  
labels <- predict(mykm,newdata)  
  
## End(Not run)
```

Index

- *Topic **clustering**
 - [dkmeans](#), 2
 - [kmeans.ddR-package](#), 2
 - [predict.dkmeans](#), 4
- *Topic **distributed clustering**
 - [dkmeans](#), 2
 - [kmeans.ddR-package](#), 2
 - [predict.dkmeans](#), 4
- *Topic **distributed kmeans**
 - [dkmeans](#), 2
 - [kmeans.ddR-package](#), 2
 - [predict.dkmeans](#), 4
- *Topic **distributed**
 - [dkmeans](#), 2
 - [kmeans.ddR-package](#), 2
 - [predict.dkmeans](#), 4
- *Topic **kmeans**
 - [dkmeans](#), 2
 - [kmeans.ddR-package](#), 2
 - [predict.dkmeans](#), 4

[dkmeans](#), 2

[fitted.dkmeans \(dkmeans\)](#), 2

[kmeans](#), 4

[kmeans.ddR \(kmeans.ddR-package\)](#), 2

[kmeans.ddR-package](#), 2

[predict.dkmeans](#), 4

[print.dkmeans \(dkmeans\)](#), 2