Package ‘yakmoR’

Type Package
Title A Simple Wrapper for the k-Means Library Yakmo
Version 0.1.1
Date 2015-08-03
Author Aydin Demircioglu
Maintainer Aydin Demircioglu <aydin.demircioglu@ini.rub.de>
Description This is a simple wrapper for the yakmo K-
    Means library (developed by Naoki Yoshinaga, see http://www.tkl.iis.u-
    tokyo.ac.jp/~ynaga/yakmo/). It performs fast and robust (orthogonal) K-Means.
Copyright Original GPL-2 C++ code by Naoki Yoshinaga (see
    http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/yakmo/)
License GPL (>= 2)
Imports Rcpp (>= 0.11.6), checkmate (>= 1.5.1), BBmisc (>= 1.9)
LinkingTo Rcpp
Suggests knitr, testthat
VignetteBuilder knitr
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-08-04 06:47:04

R topics documented:

  orthoKMeansPredict ............................................... 2
  orthoKMeansPredictCpp ......................................... 2
  orthoKMeansTrain ................................................. 3
  orthoKMeansTrainCpp ........................................... 4
  yakmoR ............................................................ 4

Index 5
**Description**

orthogonal kmeans prediction function

**Usage**

```r
orthokMeansPredict(x, obj = NULL, verbose = FALSE)
```

**Arguments**

- `x`: data to assign clusters
- `obj`: an object returned by orthoKMeansTrain
- `verbose`: show verbose messages?

**Value**

a matrix with as many columns as rounds trained

**Examples**

```r
obj <- yakmo::orthokMeanstrain(x = as.matrix(iris[seq(1,150,2),1:4]),
k = 3, rounds = 3)
predictions <- yakmo::orthokMeansPredict (x = as.matrix(iris[seq(2,150,2),1:4]),
obj = obj)
```

---

**Description**

K-Means prediction using yakmo library

**Usage**

```r
orthokMeansPredictCpp(x, centers, nf, k = 0L, verbose = FALSE)
```

**Arguments**

- `x`: data matrix
- `centers`: centers
- `nf`: number of features
- `k`: number of clusters
- `verbose`: verbose output?
orthoKMeansTrain

Value

a list consisting of cluster these are the labels for the resulting clustering, one column for each clustering.

Description

orthoKMeansTrain will cluster a given data set into the specified number of clusters. It can use either random initialization of the centroids or use KMeans++ for this. The K-Means training itself is accelerated by using techniques by Greg Hamerly. Orthoginality is implemented by using ideas from Cui et al 'Non-redundant multi-view clustering via orthogonalization'.

Usage

orthoKMeansTrain(x = NULL, k = NULL, rounds = 1, iter.max = 100,
init.type = "KMeans++", verbose = FALSE)

Arguments

x             data to cluster
k             number of centroids
rounds        number of rounds/views for orthogonal kmeans
iter.max      number of maximal iterations for each clustering
init.type     string with method to initialize centroids
verbose       show verbose messages?

Value

an S3 object containing the cluster labels for the training set as well as all necessary information for prediction.

Examples

obj = yakmo::orthoKMeansTrain (x = as.matrix(iris[seq(1,150,2),1:4]),
k = 3, rounds = 3, verbose = TRUE)
orthokMeansTrainCpp  \hspace{1em} K-Means using yakmo library

**Description**

K-Means using yakmo library

**Usage**

orthokMeansTrainCpp(x, rounds = 1L, k = 3L, iter = 100L, initType = 0L, verbose = FALSE)

**Arguments**

- **x**: data matrix
- **rounds**: number of rounds (orthogonal views)
- **k**: number of clusters
- **iter**: number of iterations in one round
- **initType**: centroid initialization via Random or KMeans++
- **verbose**: verbose output?

**Value**

- a list consisting of centers these are the resulting centroids of the kmean algorithm (as a std::vector of NumericMatrix)
- cluster these are the labels for the resulting clustering (as a std::vector of NumericVector)
- obj this is a vector with the final objective value for each round

---

yakmoR  \hspace{1em} yakmoR: A package for fast K-Means with multi-view clustering via orthogonalization.

**Description**

The yakmoR package is a wrapper for the yakmo library. It uses basic K-Means, and provides multi-view clustering via orthogonalization as well as an option to use KMeans++ initialization.

**yakmoR functions**

orthoKMeansTrain orthoKMeansPredict
Index

orthoKMeansPredict, 2
orthoKMeansPredictCpp, 2
orthoKMeansTrain, 3
orthoKMeansTrainCpp, 4

yakmoR, 4
yakmoR-package (yakmoR), 4