Package ‘MajMinKmeans’

May 7, 2026

Type Package

Title k-Means Algorithm with a Majorization-Minimization Method

Version 0.1.0

Maintainer Sheikhi Ayyub `<sheikhy.a@uk.ac.ir>`

Description A hybrid of the K-means algorithm and a Majorization-Minimization method to introduce a robust clustering. The reference paper is: Julien Mairal, (2015) <doi:10.1137/140957639>. The two most important functions in package 'MajMinKmeans' are cluster_km() and cluster_MajKm(). Cluster_km() clusters data without Majorization-Minimization and cluster_MajKm() clusters data with Majorization-Minimization method. Both of these functions calculate the sum of squares (SS) of clustering. Another useful function is MajMinOptim(), which helps to find the optimum values of the Majorization-Minimization estimator.

Imports MASS

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Sheikhi Ayyub [aut, cre],
Yaghoobi Mohammad Ali [aut]

Repository CRAN

Date/Publication 2024-05-17 09:20:09 UTC

Contents

`clusters_km` *clustering results of the k-mean algorithm*

Description

clusters data into two clusters. This functionis uses the `kmeans` function to cluster the data and exports the clustering results as well as the sum of square (SS) of clustering using the Euclidian distance.

Usage

```
clusters_km(x, k = 2)
```

Arguments

`x`	matrix of data (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`k`	number of clusters (this version considers 2 clusters)

Value

sum of square (SS) of clustring

Examples

```
{
X=rbind(matrix(rnorm(1000*2 ,4,.1),1000,2),matrix(rnorm(1000*2, 3, 0.2),1000,2))
M<- X[sample(nrow(X), 2),]
clusters_km(X,2)
}
```

`clusters_MajKm` *clustering results of the majorized k-mean algorithm*

Description

clusters data into two clusters with a majorization k-means This functionis use a hybrid of the k-means and the majorizaion-minimazation method to cluster the data and exports the clustering results as well as the sum of square (SS) of clustering

Usage

```
clusters_MajKm(X, k = 2, La)
```

Arguments

`X`	matrix of data (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`k`	number of clusters (this version considers 2 clusters)
`La`	the tunnung parameter

Value

sum of square (SS) of clustring and the 'delta' (difference of two successive majorization function).

Examples

```
{
X=rbind(matrix(rnorm(1000*2 ,4,.1),1000,2),matrix(rnorm(1000*2, 3, 0.2),1000,2))
M <- X[sample(nrow(X), 2),]
clusters_MajKm(X,2, 0.5)
}
```

`Euclid` *Euclidian distance*

Description

Calculates the Euclidian distance between points. This function can use in `kmeans` function to do the clustering procedure using the Euclidian distance.

Usage

```
Euclid(x, mu)
```

Arguments

`x`	matrix of data (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`mu`	initial seleted centroids (randomly or another method).

Value

Euclidian distance between two points.

Examples

```
{
X=rbind(matrix(rnorm(1000*2 ,4,.1),1000,2),matrix(rnorm(1000*2, 3, 0.2),1000,2))
M <- X[sample(nrow(X), 2),]
Euclid(X,M)
}
```

`kmeans` *k-means function*

Description

k-means algorithm in clustering. This function export the clustered results based on one replication of the k-means method

Usage

```
kmeans(x, centers, nItter = 4)
```

Arguments

`x`	matrix of data (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`centers`	initial seleted centroids (randomly or another method)
`nItter`	Number of itteration function

Value

clustered results based on k-means methods.

Examples

```
{
X=rbind(matrix(rnorm(1000*2 ,4,.1),1000,2),matrix(rnorm(1000*2, 3, 0.2),1000,2))
M <- X[sample(nrow(X), 2),]
kmeans(X,M, 4)
}
```

`MajMinOptim` *majorization-minimization optimization*

Description

Finding the optimized majorization-minimization centers

Usage

```
MajMinOptim(X, Z, M, eps, lambda)
```

Arguments

`X`	matrix of data (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`Z`	is a n by k matrix where for all i and j, zi,j is abinary variable that is equal to 1 if the case i is assigned to cluster j and zero otherwise. (dim 1: samples (must be equal to dim 1 of X), dim 2: attributes (must be equal to dim 2 of X))
`M`	initial seleted centroids (randomly or another method)
`eps`	a threshold value assumed as 0.0001
`lambda`	a threshold value assumed as 0.5

Value

The optimized majorization-minimization centers.

Examples

```
{
X=rbind(matrix(rnorm(1000*2 ,4,.1),1000,2),matrix(rnorm(1000*2, 3, 0.2),1000,2))
M <- X[sample(nrow(X), 2),]
distsToCenters <- Euclid(X, M)
clusters <- apply(distsToCenters, 1, which.min)
Z <- matrix(0, nrow = NROW(X), ncol = 1)
for(i in 1:NROW(X))
if (clusters[[i]] == 1)
    Z[i,]=clusters[[i]]
Z=cbind(Z, 1-Z)
MajMinOptim(X,Z,M ,eps=1e-4, lambda=.5)
}
```

Index