Package 'hkclustering'

January 13, 2018

Type Package

Title Ensemble Clustering using K Means and Hierarchical Clustering

Version 1.0.1

Date 2018-01-12

Maintainer Ilan Fridman Rojas <ilanf@profusion.com>

Description Implements an ensemble algorithm for clustering combining a k-means and a hierarchical clustering approach.

Imports cluster

License GPL-2

NeedsCompilation no

Repository CRAN

Author Kaloyan Stoyanov [aut],
Henrik Nordmark [aut],
Aris Perperoglou [aut],
Rolando Medellin [aut],
Ilan Fridman Rojas [cre],
Berthold Lausen [aut]

Date/Publication 2018-01-13 22:27:09 UTC

R topics documented:

hkclustering-package *Ensemble clustering (kmeans and hierarchical clustering) package.*

Description

Feature selection methods are often used as a preprocessing method. This package contains functions to apply an ensemble method of hierchical and kmeans clustering to a dataframe

Details

Package:	hkclustering
Type:	Package
Version:	1.0
Date:	2016-09-02
License:	GPL-2
LazyLoad:	yes

Author(s)

Kaloyan Stoyanov, Maintainer: Rolando Medellin <Rolandom@profusion.com>

References

Stoyanov,K.(2015), Hierarchicalk K-means clustering and its application in customer segmentation. Master dissertation, Department of Mathematical Sciences, University of Essex, UK.

Berthold Lausen, Kaloyan Stoyanov, Rolando Medellin, Henrik Nordmark, Aris Perperoglou, Ensemble methods for clustering and classification, 5TH GERMAN-JAPANESE WORKSHOP ON CLASSIFICATION, 2016.

centroidssummary *Returns Centroids summary*

Description

Returns Centroid summary

Usage

```
centroidssummary(clustereddata)
```

Arguments

clustereddata	original dataframe

Value

centroid

Author(s)

Kaloyan S, <kaloyanS@profusion.com>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (clustereddata)
{
    colnames(clustereddata)[(length(clustereddata))] <- "cluster_number"
    centroids <- aggregate(clustereddata, by = list(clustereddata$cluster_number),
        FUN = mean)
    clustereddata$counts <- 1
    centroids <- cbind(centroids, aggregate(counts ~ cluster_number,
        data = clustereddata, FUN = sum))
    centroids <- centroids[, c((length(df) + 2), 2:(length(df) +
        1), (length(df) + 4))]
    return(centroids)
  }
```

---

hkclusplus *hkclusplus*

Description

Takes a dataframe and performs kmeans and a hierarchical clustering on the dataframe using teh gap statistic to calculate the initial number of centroids. The function outputs a dataframe as the clustered data

Usage

```
hkclusplus(df, t)
```

Arguments

df	original dataframe
t	Number of iterations to find the centroids

Author(s)

Kaloyan S <kaloyanS@profusion.com>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

a<-runif(300, min=3.5, max=2000)
b<-runif(300, min=1.5, max=2000)
df = data.frame(a, b)

#Let the Gap statistic to find the clusters
results.hkplus<-hkclusplus(df,100)
centroidssummary(results.hkplus)
with(results.hkplus, pairs(results.hkplus[,1:2], col=c(1:7)[results.hkplus[,3]]))

## The function is currently defined as
function (df, t)
{
    library(cluster)
    scaled.df <- scale(df)
    numbk <- which.max(clusGap(scaled.df, FUN = kmeans, K.max = 8,
        B = 200)$Tab[, 3])
    rm(.Random.seed, envir = globalenv())
    temp <- kmeans(scaled.df, numbk)
    c <- temp$centers
    c <- temp$centers
    for (i in 2:t) {
        rm(.Random.seed, envir = globalenv())
        temp <- kmeans(scaled.df, numbk)
        c <- rbind(c, temp$centers)
    }
    cr <- as.data.frame(c, row.names = F)
    d <- dist(cr, method = "euclidean")
    fit <- hclust(d, method = "centroid")
    cr$clusnumber <- cutree(fit, k = numbk)
    centroids1 <- aggregate(cr, by = list(cr$clusnumber), FUN = mean)
    centr <- centroids1[, c(2:(length(df) + 1))]
    final <- kmeans(scaled.df, centr)
    clustereddata <- cbind(df, final$cluster)
    colnames(clustereddata)[(length(df) + 1)] <- "cluster_number"
    return(clustereddata)
  }
```

---

hkclustering *Clustering function*

Description

Takes a dataframe and the number of initial clusters and performs kmeans and a hierarchical clustering on the dataframe. The function outputs a dataframe as the clustered data

Usage

```
hkclustering(df, numbk, t)
```

Arguments

df	Original dataframe to cluster
numbk	The number of initial clusters for the kmeans algorithm
t	Number of iterations to find the centroids

Author(s)

Kaloyan S, <kaloyanS@profusion.com>

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.


a<-runif(500, min=3.5, max=2000)
b<-runif(500, min=1.5, max=2000)
df = data.frame(a, b)

#Specifying  4 clusters
results.hkclust<-hkclustering(df,4,100)
centroidssummary(results.hkclust)
with(results.hkclust, pairs(results.hkclust[,1:2], col=c(1:10)[results.hkclust[,3]]))

## The function is currently defined as
function (df, numbk, t)
{
    scaled.df <- scale(df)
    rm(.Random.seed, envir = globalenv())
    temp <- kmeans(scaled.df, numbk)
    c <- temp$centers
    c <- temp$centers
    for (i in 2:t) {
        rm(.Random.seed, envir = globalenv())
        temp <- kmeans(scaled.df, numbk)
        c <- rbind(c, temp$centers)
    }
    cr <- as.data.frame(c, row.names = F)
    d <- dist(cr, method = "euclidean")
    fit <- hclust(d, method = "centroid")
    cr$clusnumber <- cutree(fit, k = numbk)
```

```
  centroids1 <- aggregate(cr, by = list(cr$clusnumber), FUN = mean)
  centr <- centroids1[, c(2:(length(df) + 1))]
  final <- kmeans(scaled.df, centr)
  clustereddata <- cbind(df, final$cluster)
  colnames(clustereddata)[(length(df) + 1)] <- "cluster_number"
  return(clustereddata)
}
```

Index