Package ‘MCSim’

October 12, 2022

Type Package
Title Determine the Optimal Number of Clusters
Version 1.0
Date 2018-09-30
Author Ahmed N. Albatineh, Meredith L. Wilcox, Bashar Zogheib, Magdalena Niewiadomska-Bugaj
Maintainer Ahmed N. Albatineh <aalbatineh@hsc.edu.kw>
Description Identifies the optimal number of clusters by calculating the similarity between
two clustering methods at the same number of clusters using the corrected indices of Rand and
Jaccard as described in Albatineh and Niewiadomska-Bugaj (2011). The number of clusters at
which the index attain its maximum more frequently is a candidate for being the optimal
number of clusters.
Depends R (>= 3.1.0)
Imports MASS,CircStats,stats,graphics
License GPL-2
Encoding UTF-8
LazyData true
NeedsCompilation no
Repository CRAN
Date/Publication 2018-10-08 11:00:17 UTC

R topics documented:

MCS ................................................ .................................................. 2

Index 5

1
**MCSim: a Package to Determine the Optimal Number of Clusters**

**Description**

This package identifies the optimal number of clusters by calculating the similarity between two clustering methods at the same number of clusters using the corrected indices of Rand and Jaccard as described in Albatineh and Niewiadomska-Bugaj (2011). The number of clusters at which the index attain its maximum more frequently is a candidate for being the optimal number of clusters.

**Usage**

MCS(data1=data1, nc=nc, method1="method1", method2="method2", index="index", print.stats=FALSE, st.data=FALSE, plot.hc=FALSE, circ=FALSE, convert=TRUE, plot.data=FALSE)

**Arguments**

- **data1**: Numeric data matrix to be clustered.
- **nc**: Maximum number of clusters, similarity will be calculated for 2<= nc < n-1.
- **method1**: First clustering method to be used. One of "single","average","complete","ward" ,"median","mcquitty","kmeans")
- **method2**: Second clustering method to be used. One of "single","average","complete","ward","median","mcquitty",
- **index**: Similarity index to be used. Either "rand" or "jaccard" index which will be corrected for chance agreement.
- **print.stats**: Logical, if "TRUE" the similarity will be outputed for each value between 2 and nc.
- **st.data**: Logical, if "TRUE" data will be standardzed. This is for linear (non-circular) data only.
- **plot.hc**: Logical, if "TRUE" hierarchical clustering tree (dendrogram) will be produced. This is for linear (non-circular) data only.
- **circ**: Logical, if "TRUE" data are circular or measured as angles.
- **convert**: Logical, if "TRUE" data will be converted from angular to radians. This is for circular data only.
- **plot.data**: Logical, if "TRUE" a circular plot of the data will be produced. This is for circular data only.

**Details**

The distance measure used to calculate the distance for linear data is the Euclidean distance. For circular data the distance is calculated using the formula dij=0.5*(1 - cos(Aii - Bjj)). The correction for Rand index is based on the expectation by Hubert and Arabie (1985). For correcting Jaccard index, see Albatineh & Niewiadomska-Bugaj (2011).
Value

Similarity between the two clustering algorithms at each value of nc will be calculated, where 2 <= nc < n - 1, and a plot of the number of clusters vs. the value of either similarity index rand or jaccard will be produced.

Note

The following packages are needed: "MASS","CircStats","stats","datasets","graphics"

Author(s)

Ahmed N. Albatineh, Meredith L. Wilcox, Bashar Zogheib, Magdalena Niewiadomska-Bugaj

References


Examples

library("MASS")
library("CircStats")
library("stats")
library("datasets")
library("graphics")

########### Simulated data from four bivariate normal distributions
set.seed(12345)
clust1 <- mvrnorm(100,mu=c(5,5),Sigma=matrix(c(1,0.5,0.5,1),ncol=2))
clust2 <- mvrnorm(100,mu=c(5,20),Sigma=matrix(c(1,0.5,0.5,1),ncol=2))
clust3 <- mvrnorm(100,mu=c(20,5),Sigma=matrix(c(1,0.5,0.5,1),ncol=2))
clust4 <- mvrnorm(100,mu=c(20,20),Sigma=matrix(c(1,0.5,0.5,1),ncol=2))
simdat <- rbind(clust1,clust2,clust3,clust4)

MCS(data1=simdat, nc=10, method1="single", method2="ward.D2", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)

MCS(data1=simdat, nc=10, method1="kmeans", method2="single", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)

#################################
## Data from three bivariate normal distributions (elongated clusters)
set.seed(1965)
clust1 <- mvrnorm(100,mu=c(5,5),Sigma=matrix(c(1,0.9,0.9,1),ncol=2))
clust2 <- mvrnorm(100,mu=c(5,20),Sigma=matrix(c(1,0.9,0.9,1),ncol=2))

MCS(data1=simdat, nc=10, method1="single", method2="ward.D2", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)

MCS(data1=simdat, nc=10, method1="kmeans", method2="single", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)
clust3<- mvrnorm(100,mu=c(20,5),Sigma=matrix(c(1,0.9,0.9,1),ncol=2))
simdat<- rbind(clust1,clust2,clust3)
MCS(data1=simdat, nc=10, method1="complete", method2="average", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)
MCS(data1=simdat, nc=10, method1="median", method2="kmeans", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)

# Old Faithful Geyser Data Example
library("datasets")
data1<- as.matrix(faithful,nrows=272,ncol=2,byrows=TRUE)
MCS(data1=data1, nc=10, method1="average", method2="ward.D2", index="rand", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)
MCS(data1=data1, nc=10, method1="average", method2="kmeans", index="jaccard", print.stats=TRUE, st.data=FALSE, plot.hc=FALSE)

## Simulated Circular data from five von Mises distributions ####
set.seed(1945)
clust1<- as.vector(rvm(50,5,15))
clust2<- as.vector(rvm(50,10,15))
clust3<- as.vector(rvm(50,15,15))
clust4<- as.vector(rvm(50,20,15))
clust5<- as.vector(rvm(50,25,15))
data1<- rbind(clust1,clust2,clust3,clust4,clust5)
MCS(data1=data1, nc=10, method1="kmeans", method2="complete", index="rand", print.stats=TRUE, circ=TRUE, convert=FALSE, plot.data=FALSE)

### Turtles Data Example
 turtles<- c(8,9,13,13,14,18,22,27,30,34,
 38,38,40,44,45,47,48,48,48,48,48,50,53,56,
 57,58,58,61,63,64,64,64,65,65,68,70,73,
 78,78,83,83,88,88,88,90,92,92,93,95,
 96,98,100,103,106,113,118,138,153,153,
 155,204,215,223,226,237,238,243,244,245,250,
 251,257,268,285,319,343,350)
MCS(data1=turtles, nc=10, method1="single", method2="ward.D2", index="rand", print.stats=TRUE, circ=TRUE, convert=TRUE, plot.data=FALSE)
MCS(data1=turtles, nc=10, method1="ward.D2", method2="kmeans", index="jaccard", print.stats=TRUE, circ=TRUE, convert=TRUE, plot.data=FALSE)

##### Wind data example ####
 wind<- c(67,87,101,101,103,131,140,142,144,149,182,
 199,206,251,253,278,279,287,290,295,299,301,301,308,308,
 308,310,312,316,319,319,325,325,326,331,334,15)
MCS(data1=wind, nc=10, method1="ward.D2", method2="median", index="jaccard", print.stats=TRUE, circ=TRUE, convert=TRUE, plot.data=FALSE)
MCS(data1=wind, nc=10, method1="complete", method2="average", index="jaccard", print.stats=TRUE, circ=TRUE, convert=TRUE, plot.data=FALSE)
Index

- Clustering algorithm
  MCS, 2
- Correction for chance agreement
  MCS, 2
- Jaccard index
  MCS, 2
- Number of clusters
  MCS, 2
- Rand index
  MCS, 2
- Similarity index
  MCS, 2
- Validity index
  MCS, 2

MCS, 2