Package 'MKMeans'

August 20, 2025

Type Package

Title A Modern K-Means (MKMeans) Clustering Algorithm

Version 3.2

Date 2025-08-20

Depends methods, MASS

Description
It's a Modern K-Means clustering algorithm which works for data of any number of dimensions, has no limit with number of clusters expected, and can start with any initial cluster centers.

Collate AllClasses.R MKMeans.R C.f.R Dist.R

License GPL-2

NeedsCompilation no

Author Yarong Yang [aut, cre],
Nader Ebrahimi [ctb],
Yoram Rubin [ctb],
Jacob Zhang [ctb]

Maintainer Yarong Yang <`Yi.YA_yaya@hotmail.com`>

Repository CRAN

Date/Publication 2025-08-20 14:40:07 UTC

Contents

`MKMeans-package` *Modern K-Means (MKMeans) Clustering.*

Description

It's a Modern K-Means clustering algorithm which works for data of any number of dimensions, has no limit with the number of clusters expected, and can start with any initial cluster centers.

Details

Package:	MKMeans
Type:	Package
Version:	3.2
Date:	2025-08-20
License:	GPL-2

Author(s)

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang

References

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang.(2025) MKMeans: A Modern K-Means Clustering Algorithm. technical report

Examples

```
# Example 1:

# Generate 20 bivarate samples
x<-rnorm(20,0,1)
y<-rnorm(20,1,1)
data.test<-cbind(x,y)

# Conduct MKMeans analysis with K=3 and taking the first 3 samples as initial cluster centers
Res<-MKMeans(data.test,3,1,iteration=1000,tol=.95,type=1)
Ress<-Res
names(Ress@Classes[[1]])<-rep("red",length(Res@Classes[[1]]))
names(Ress@Classes[[2]])<-rep("blue",length(Res@Classes[[2]]))
names(Ress@Classes[[3]])<-rep("green",length(Res@Classes[[3]]))
Cols<-names(sort(c(Ress@Classes[[1]],Ress@Classes[[2]],Ress@Classes[[3]])))
plot(x,y,type="p",col=Cols,lwd=2)
points(Res@Centers,pch=15,col=c("red","blue","green"))

# Example 2:
```

```
library(MASS)
# Generate 10 bivariate normal samples
mu1 <- c(0, 0)
sigma1 <- matrix(c(1, 0.5, 0.5, 1), nrow=2)
SP1 <- mvrnorm(n=10, mu=mu1, Sigma=sigma1)

# Generate another 10 bivariate normal samples
mu2<-c(1,1)
sigma2<-matrix(c(1,0,0,1),nrow=2)
SP2<-mvrnorm(n=10,mu=mu2,Sigma=sigma2)

# Generate 10 more new bivariate normal samples
mu3<-c(2,2)
sigma3<-matrix(c(1,0.5,0.5,1),nrow=2)
SP3<-mvrnorm(n=10,mu=mu3,Sigma=sigma3)

# Combine the three groups of bivariate normal samples
data<-rbind(SP1,SP2,SP3)

# Conduct MKMeans analysis with K=4 and randomly picking four samples as initial cluster centers
Res<-MKMeans(data,4,data[sample(1:30,4),],iteration=1000,tol=.95,type=1)
names(Res@Classes[[1]])<-rep("red",length(Res@Classes[[1]]))
names(Res@Classes[[2]])<-rep("blue",length(Res@Classes[[2]]))
names(Res@Classes[[3]])<-rep("green",length(Res@Classes[[3]]))
names(Res@Classes[[4]])<-rep("black",length(Res@Classes[[4]]))
Cols<-names(sort(c(Res@Classes[[1]],Res@Classes[[2]],Res@Classes[[3]],Res@Classes[[4]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab="",ylab="")
points(Res@Centers,pch=5,col=c("red","blue","green","black"))
```

`C.f` *Finding the center of a cluster.*

Description

It's a function of finding the center of a cluster.

Usage

```
C.f(dat, type)
```

Arguments

`dat`	Numeric. A cluster matrix with each row being an observaion.
`type`	Integer. The type of distance between observations. 1 for Euclidean distance. 2 for Manhattan distance. 3 for maximum deviation along dimensions.

Value

A vector.

Author(s)

Yarong Yang

Examples

```
x<-rnorm(5,0,1)
y<-rnorm(5,1,1)
data<-cbind(x,y)
Res<-C.f(dat=data,type=1)
```

`Dist` *Finding the distance between two observations.*

Description

It's a function of finding the distance between two observations.

Usage

```
Dist(x,y,type)
```

Arguments

`x`	Numeric. A vector denoting an observation.
`y`	Numeric. A vector denoting an observation.
`type`	Integer. The type of distance between observations. 1 for Euclidean distance. 2 for Manhattan distance. 3 for maximum deviation among dimensions.

Value

A numeric number.

Examples

```
x<-rnorm(10,0,1)
y<-rnorm(10,1,1)
z<-rnorm(10,2,1)
data<-cbind(x,y,z)
Res<-Dist(data[1,],data[2,],type=1)
```

`MKMean` *Class to contain the results from function MKMeans.*

Description

The function MKMeans return object of class MKMean that contains the number of clusters, the center of each cluster, and the observations in each cluster.

Objects from the Class

new("MKMean",K=new("numeric"),Centers=new("matrix"),Classes=new("list"),Clusters=new("list"))

Slots

`K`**:** An integer being the number of clusters.

`Centers`**:** A numeric matrix with each row being center of a cluster.

`Classes`**:** An integer list showing the original indexes of the observations in each cluster.

`Clusters`**:** A numeric list showing the observations in each cluster.

Author(s)

Yarong Yang

References

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang.(2025) MKMeans: A Modern K-Means Clustering Algorithm. technical report

Examples

```
showClass("MKMean")
```

`MKMeans` *Modern K-Means clustering.*

Description

It's a Modern K-Means clustering algorithm which works for data of any number of dimensions, has no limit with the number of clusters expected, and can start with any initial cluster centers.

Usage

```
MKMeans(data, K, initial, iteration, tol, type)
```

Arguments

`data`	Numeric. An observation matrix with each row being an oberservation.
`K`	Integer. The number of clusters expected.
`initial`	Numeric. Either the selected initial center matrix with each row being an observation, or 1 for the first K rows of the data matrix being the intial center.
`iteration`	Integer. The number of the most iterations wanted for the clustering process.
`tol`	Numeric. The minimum acceptable percentage of stable observations to stop the clustering process, basically greater than 0.5 to guarantee the value of the results.
`type`	Integer. The type of distance between observations. 1 for Euclidean distance. 2 for Manhattan distance. 3 for maximum deviation among dimensions.

Value

An object of class MKMean.

Author(s)

Yarong Yang

References

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang.(2025) MKMeans: A Modern K-Means Clustering Algorithm. technical report

Examples

```
library(MASS)
# Generate 10 bivariate normal samples
mu1 <- c(0, 0)
sigma1 <- matrix(c(1, 0.5, 0.5, 1), nrow=2)
SP1 <- mvrnorm(n=10, mu=mu1, Sigma=sigma1)

# Generate another 10 bivariate normal samples
mu2<-c(1,1)
sigma2<-matrix(c(1,0,0,1),nrow=2)
SP2<-mvrnorm(n=10,mu=mu2,Sigma=sigma2)

# Generate 10 more new bivariate normal samples
mu3<-c(2,2)
sigma3<-matrix(c(1,0.5,0.5,1),nrow=2)
SP3<-mvrnorm(n=10,mu=mu3,Sigma=sigma3)

# Combine the three groups of bivariate normal samples
data<-rbind(SP1,SP2,SP3)

# Conduct MKMeans analysis with K=3 and randomly picking three samples as initial cluster centers
Res<-MKMeans(data,3,data[sample(1:30,3),],iteration=1000,tol=.95,type=1)
names(Res@Classes[[1]])<-rep("red",length(Res@Classes[[1]]))
```

```
names(Res@Classes[[2]])<-rep("blue",length(Res@Classes[[2]]))
names(Res@Classes[[3]])<-rep("green",length(Res@Classes[[3]]))
Cols<-names(sort(c(Res@Classes[[1]],Res@Classes[[2]],Res@Classes[[3]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab="",ylab="")
points(Res@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results with the original samples
par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=rep(c("sky blue","orange","purple"),rep(10,3)),
     lwd=2,xlab="",ylab="",main="Original Data")
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab="",ylab="",
     main="MKMeans Clustering Results")
points(Res@Centers,pch=5,col=c("red","blue","green"))

# conduct MKMeans analysis with K=4 and randomly picking four samples as initial cluster centers
Res<-MKMeans(data,4,data[sample(1:30,4),],iteration=1000,tol=.95,type=1)
names(Res@Classes[[1]])<-rep("red",length(Res@Classes[[1]]))
names(Res@Classes[[2]])<-rep("blue",length(Res@Classes[[2]]))
names(Res@Classes[[3]])<-rep("green",length(Res@Classes[[3]]))
names(Res@Classes[[4]])<-rep("black",length(Res@Classes[[4]]))
Cols<-names(sort(c(Res@Classes[[1]],Res@Classes[[2]],Res@Classes[[3]],Res@Classes[[4]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab="",ylab="")
points(Res@Centers,pch=5,col=c("red","blue","green","black"))

#  Compare the clustering results with the original data
par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=rep(c("sky blue","orange","purple"),rep(10,3)),
     lwd=2,xlab="",ylab="",main="Original Data")
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab="",ylab="",
     main="MKMeans Clustering Results")
points(Res@Centers,pch=5,col=c("red","blue","green","black"))
```

Index