Package ‘KQM’

October 16, 2025

Type Package

Title K Quantiles Medoids (KQM) Clustering

Version 1.1.1

Date 2025-10-13

Depends methods, MASS, gtools, cluster

Description K Quantiles Medoids (KQM) clustering applies quantiles to divide data of each dimension into K mean intervals. Combining quantiles of all the dimensions of the data and fully permuting quantiles on each dimension is the strategy to determine a pool of candidate initial cluster centers. To find the best initial cluster centers from the pool of candidate initial cluster centers, two methods based on quantile strategy and PAM strategy respectively are proposed. During a clustering process, medoids of clusters are used to update cluster centers in each iteration. Comparison between KQM and the method of randomly selecting initial cluster centers shows that KQM is almost always getting clustering results with smaller total sum squares of distances.

Collate AllClasses.R KQM.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 2025-10-16 18:30:19 UTC

Contents

KQM-package *K Quantiles Medoids (KQM) Clustering.*

Description

K Quantiles Medoids (KQM) clustering applies quantiles to divide data of each dimension into K mean intervals. Combining quantiles of all the dimensions of the data and fully permuting quantiles on each dimension is the strategy to determine a pool of candidate initial cluster centers. To find the best initial cluster centers from the pool of candidate initial cluster centers, two methods based on quantile stategy and PAM strategy respectively are proposed. During a clustering process, medoids of clusters are used to update cluster centers in each iteration. Comparison between KQM and the method of randomly selecting initial cluster centers shows that KQM is almost always getting clustering results with smaller total sum squares of distances.

Details

Package:	KQM
Type:	Package
Version:	1.1.1
Date:	2025-10-13
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) KQM: K Quantiles Medoids (KQM) Clustering. technical report in preparation

Examples

```
# Generate 10 bivariate normal samples
require(MASS)
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 KQM analysis with K=3 by the "YY" method and quantile strategy is
# selected to determine the initial cluster centers
Res<-KQM(data,3,method="YY",iteration=1000,type=1,style=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=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results with the original samples
oldpar<-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=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
on.exit<-par(oldpar)

# Conduct KQM analysis with K=3 and randomly picking 3 samples as initial
#  cluster centers
Res2<-KQM(data,3,method="initial",initial=data[sample(1:nrow(data),3),],iteration=1000,type=1)
names(Res2@Classes[[1]])<-rep("red",length(Res2@Classes[[1]]))
names(Res2@Classes[[2]])<-rep("blue",length(Res2@Classes[[2]]))
names(Res2@Classes[[3]])<-rep("green",length(Res2@Classes[[3]]))
Cols2<-names(sort(c(Res2@Classes[[1]],Res2@Classes[[2]],Res2@Classes[[3]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols2,lwd=2,xlab=paste("Total SSE = ",
     round(Res2@SSE[length(Res2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'initial' method")
points(Res2@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results by the above "YY" method and the "initial" method
oldpar<-par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
plot(data[,1],data[,2],type="p",pch=19,col=Cols2,lwd=2,xlab=paste("Total SSE = ",
     round(Res2@SSE[length(Res2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'initial' method")
points(Res2@Centers,pch=5,col=c("red","blue","green"))
```

```
on.exit(oldpar)

# Conduct KQM analysis with K=3 by the "YY" method and PAM strategy is selected to
# determine the initial cluster centers
Res.2<-KQM(data,3,method="YY",iteration=1000,type=1,style=2)
names(Res.2@Classes[[1]])<-rep("red",length(Res.2@Classes[[1]]))
names(Res.2@Classes[[2]])<-rep("blue",length(Res.2@Classes[[2]]))
names(Res.2@Classes[[3]])<-rep("green",length(Res.2@Classes[[3]]))
Cols.2<-names(sort(c(Res.2@Classes[[1]],Res.2@Classes[[2]],Res.2@Classes[[3]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols.2,lwd=2,xlab=paste("Total SSE = ",
     round(Res.2@SSE[length(Res.2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 2")
points(Res.2@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results by the "YY" method with style 1 and style 2 respectively
oldpar<-par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, YY method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
plot(data[,1],data[,2],type="p",pch=19,col=Cols.2,lwd=2,xlab=paste("Total SSE = ",
     round(Res.2@SSE[length(Res.2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 2")
points(Res.2@Centers,pch=5,col=c("red","blue","green"))
on.exit<-par(oldpar)
```

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

Description

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

Usage

```
C.f(DM, Cl.ind)
```

Arguments

DM	Numeric. The distance matrix with the element of the i-th row and the j-th column being the distance between the i-th object and the j-th object.
Cl.ind	Integer. The index vector of a cluster.

Value

A vector.

Author(s)

Yarong Yang

Examples

```
### Generate a distance matrix with 10 observations
dm1<-matrix(rnorm(100,0,1),10,10)
dm2<-t(dm1)
dm<-dm1+dm2
for(i in 1:10)
     dm[i,i]<-0
DM<-abs(dm)

### Randomly generate a cluster with 5 samples
Cl.ind<-sample(1:10,5)

Res<-C.f(DM,Cl.ind)
```

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)
```

KQM *K Quantiles Medoids (KQM) Clustering.*

Description

K Quantiles Medoids (KQM) clustering applies quantiles to divide data of each dimension into K mean intervals. Combining quantiles of all the dimensions of the data and fully permuting quantiles on each dimension is the strategy to determine a pool of candidate initial cluster centers. To find the best initial cluster centers from the pool of candidate initial cluster centers, two methods based on quantile stategy and PAM strategy respectively are proposed. During a clustering process, medoids of clusters are used to update cluster centers in each iteration. Comparison between KQM and the method of randomly selecting initial cluster centers shows that KQM is almost always getting clustering results with smaller total sum squares of distances.

Usage

```
KQM(data, K, method, initial, iteration, type, style)
```

Arguments

`data`	Numeric. An observation matrix with each row being an oberservation.
`K`	Integer. The number of clusters expected.
`method`	Character. "YY" or "initial". No initial cluster centers are required for "YY" method. "initial" method can work for any initial cluster centers.
`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.
`type`	Integer. The type of distance between observations. 1 for Euclidean distance. 2 for Manhattan distance. 3 for maximum deviation among dimensions.
`style`	Integer. 1 or 2. 1 for quantile strategy and 2 for PAM strategy respectively in the stage of selecting initial cluster centers.

Value

An object of class KQMobj.

Author(s)

Yarong Yang

References

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang.(2025) KQM: K Quantiles Medoids (KQM) Clustering. technical report in preparation

Examples

```
# Generate 10 bivariate normal samples
require(MASS)
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 KQM analysis with K=3 by the "YY" method and quantile strategy is selected
# to determine the initial cluster centers
Res<-KQM(data,3,method="YY",iteration=1000,type=1,style=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=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results with the original samples
oldpar<-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=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
on.exit<-par(oldpar)

# Conduct KQM analysis with K=3 and randomly picking 3 samples as initial
# cluster centers
Res2<-KQM(data,3,method="initial",initial=data[sample(1:nrow(data),3),],iteration=1000,type=1)
names(Res2@Classes[[1]])<-rep("red",length(Res2@Classes[[1]]))
names(Res2@Classes[[2]])<-rep("blue",length(Res2@Classes[[2]]))
names(Res2@Classes[[3]])<-rep("green",length(Res2@Classes[[3]]))
Cols2<-names(sort(c(Res2@Classes[[1]],Res2@Classes[[2]],Res2@Classes[[3]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols2,lwd=2,xlab=paste("Total SSE = ",
     round(Res2@SSE[length(Res2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'initial' method")
```

```
points(Res2@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results by the above "YY" method and the "initial" method
oldpar<-par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
plot(data[,1],data[,2],type="p",pch=19,col=Cols2,lwd=2,xlab=paste("Total SSE = ",
     round(Res2@SSE[length(Res2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'initial' method")
points(Res2@Centers,pch=5,col=c("red","blue","green"))
on.exit(oldpar)

# Conduct KQM analysis with K=3 by the "YY" method and PAM strategy is selected to
# determine the initial cluster centers
Res.2<-KQM(data,3,method="YY",iteration=1000,type=1,style=2)
names(Res.2@Classes[[1]])<-rep("red",length(Res.2@Classes[[1]]))
names(Res.2@Classes[[2]])<-rep("blue",length(Res.2@Classes[[2]]))
names(Res.2@Classes[[3]])<-rep("green",length(Res.2@Classes[[3]]))
Cols.2<-names(sort(c(Res.2@Classes[[1]],Res.2@Classes[[2]],Res.2@Classes[[3]])))
plot(data[,1],data[,2],type="p",pch=19,col=Cols.2,lwd=2,xlab=paste("Total SSE = ",
     round(Res.2@SSE[length(Res.2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 2")
points(Res.2@Centers,pch=5,col=c("red","blue","green"))

#  Compare the clustering results by the "YY" method with style 1 and style 2 respectively
oldpar<-par(mfrow=c(1,2))
plot(data[,1],data[,2],type="p",pch=19,col=Cols,lwd=2,xlab=paste("Total SSE = ",
     round(Res@SSE[length(Res@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, YY method, style 1")
points(Res@Centers,pch=5,col=c("red","blue","green"))
plot(data[,1],data[,2],type="p",pch=19,col=Cols.2,lwd=2,xlab=paste("Total SSE = ",
     round(Res.2@SSE[length(Res.2@SSE)],2),sep=""),ylab="",
     main="KQM Clustering Results, 'YY' method, style 2")
points(Res.2@Centers,pch=5,col=c("red","blue","green"))
on.exit<-par(oldpar)
```

KQMobj *Class to contain the results from function KQM.*

Description

The function KQM returns object of class KQMobj that contains the number of clusters, centers of clusters, positions of observations in each cluster, the observations in each cluster, and the sum squares of distances in each cluster and all the clusters.

Objects from the Class

new("KQMobj",K=new("numeric"),DM=new("matrix"),Centers.ix=new("vector"),Centers=new("matrix"),Classes=new("lis

Slots

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

`DM`**:** A numeric matrix with the element of the i-th row and the j-th column being the distance between the i-th object and the j-th object.

`Centers.ix`**:** An integer vector showing the original positions of the cluster centers.

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

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

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

`SSE`**:** A numeric vector composed of SSE of each cluster and the total SSE of all the clusters.

Author(s)

Yarong Yang

References

Yarong Yang, Nader Ebrahimi, Yoram Rubin, and Jacob Zhang.(2025) K Quantiles Medoids (KQM) Clustering. technical report in preparation

Examples

```
showClass("KQMobj")
```

Index