

# Package ‘trimcluster’

February 9, 2020

**Title** Cluster Analysis with Trimming

**Version** 0.1-5

**Date** 2020-02-08

**VersionNote** Released 0.1-2.1 on 2018-07-20 on CRAN

**Author** Christian Hennig <chrish@stats.ucl.ac.uk>

**Depends** R (>= 1.9.0)

**Suggests** fpc

**Description** Trimmed k-means clustering. The method is described in Cuesta-Albertos et al. (1997) <doi:10.1214/aos/1031833664>.

**Maintainer** Valentin Todorov <valentin.todorov@chello.at>

**License** GPL

**URL** <http://www.homepages.ucl.ac.uk/~ucakche/>

**Repository** CRAN

**Date/Publication** 2020-02-09 12:10:02 UTC

**NeedsCompilation** no

## R topics documented:

trimkmeans . . . . .	1
<b>Index</b>	<b>4</b>

---

trimkmeans	<i>Trimmed k-means clustering</i>
------------	-----------------------------------

---

### Description

The trimmed k-means clustering method by Cuesta-Albertos, Gordaliza and Matran (1997). This optimizes the k-means criterion under trimming a portion of the points.

**Usage**

```

trimkmeans(data,k,trim=0.1, scaling=FALSE, runs=100, points=NULL,
            countmode=runs+1, printcrit=FALSE,
            maxit=2*nrow(as.matrix(data)))

## S3 method for class 'tkm'
print(x, ...)
## S3 method for class 'tkm'
plot(x, data, ...)

```

**Arguments**

<code>data</code>	matrix or data.frame with raw data
<code>k</code>	integer. Number of clusters.
<code>trim</code>	numeric between 0 and 1. Proportion of points to be trimmed.
<code>scaling</code>	logical. If TRUE, the variables are centered at their means and scaled to unit variance before execution.
<code>runs</code>	integer. Number of algorithm runs from initial means (randomly chosen from the data points).
<code>points</code>	NULL or a matrix with k vectors used as means to initialize the algorithm. If initial mean vectors are specified, runs should be 1 (otherwise the same initial means are used for all runs).
<code>countmode</code>	optional positive integer. Every countmode algorithm runs trimkmeans shows a message.
<code>printcrit</code>	logical. If TRUE, all criterion values (mean squares) of the algorithm runs are printed.
<code>maxit</code>	integer. Maximum number of iterations within an algorithm run. Each iteration determines all points which are closer to a different cluster center than the one to which they are currently assigned. The algorithm terminates if no more points have to be reassigned, or if maxit is reached.
<code>x</code>	object of class tkm.
<code>...</code>	further arguments to be transferred to plot or <a href="#">plotcluster</a> .

**Details**

plot.tkm calls [plotcluster](#) if the dimensionality of the data p is 1, shows a scatterplot with non-trimmed regions if p=2 and discriminant coordinates computed from the clusters (ignoring the trimmed points) if p>2.

**Value**

An object of class 'tkm' which is a LIST with components

<code>classification</code>	integer vector coding cluster membership with trimmed observations coded as k+1.
<code>means</code>	numerical matrix giving the mean vectors of the k classes.

disttom	vector of squared Euclidean distances of all points to the closest mean.
ropt	maximum value of disttom so that the corresponding point is not trimmed.
k	see above.
trim	see above.
runs	see above.
scaling	see above.

**Author(s)**

Christian Hennig <chris@stats.ucl.ac.uk> <http://www.homepages.ucl.ac.uk/~ucakche/>

**References**

Cuesta-Albertos, J. A., Gordaliza, A., and Matran, C. (1997) Trimmed k-Means: An Attempt to Robustify Quantizers, *Annals of Statistics*, 25, 553-576.

**See Also**

[plotcluster](#)

**Examples**

```

set.seed(10001)
n1 <-60
n2 <-60
n3 <-70
n0 <-10
nn <- n1+n2+n3+n0
pp <- 2
X <- matrix(rep(0,nn*pp),nrow=nn)
ii <-0
for (i in 1:n1){
  ii <-ii+1
  X[ii,] <- c(5,-5)+rnorm(2)
}
for (i in 1:n2){
  ii <- ii+1
  X[ii,] <- c(5,5)+rnorm(2)*0.75
}
for (i in 1:n3){
  ii <- ii+1
  X[ii,] <- c(-5,-5)+rnorm(2)*0.75
}
for (i in 1:n0){
  ii <- ii+1
  X[ii,] <- rnorm(2)*8
}
tkm1 <- trimkmeans(X,k=3,trim=0.1,runs=3)
# runs=3 is used to save computing time.
print(tkm1)
plot(tkm1,X)

```

# Index

\*Topic **cluster**

trimkmeans, 1

\*Topic **multivariate**

trimkmeans, 1

plot.tkm(trimkmeans), 1

plotcluster, 2, 3

print.tkm(trimkmeans), 1

trimkmeans, 1