Package 'briKmeans'

February 15, 2021

Version 0.1

Date 2021-01-30

Title Package for Brik and Fabrik Algorithms to Initialise Kmeans

Author Javier Albert Smet <javas@kth.se> and
Aurora Torrente <etorrent@est-econ.uc3m.es>.

Maintainer Aurora Torrente <etorrent@est-econ.uc3m.es>

Depends R (>= 3.1.0), boot, cluster, depthTools

Description Implementation of the BRIk and FABRIk algorithms to initialise k-means. These methods are intended for the clustering of multivariate and functional data, respectively. They make use of the Modified Band Depth and bootstrap to identify appropriate initial seeds for k-means, which are proven to be better options than many techniques in the literature. Torrente and Romo (2020) <doi:10.1007/s00357-020-09372-3>.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-15 09:40:10 UTC

R topics documented:

brik *Computation of Initial Seeds and Kmeans Results*

Description

brik computes appropriate seeds –based on bootstrap and the MBD depth– to initialise k-means, which is then run.

Usage

```
brik(x, k, method="Ward", nstart=1, B=10, J = 2, ...)
```

Arguments

x	a data matrix containing N observations (individuals) by rows and d variables (features) by columns
k	number of clusters
method	clustering algorithm used to cluster the cluster centres from the bootstrapped replicates; Ward, by default. Currently, only pam and randomly initialised kmeans are implemented
nstart	number of random initialisations when using the kmeans method to cluster the cluster centres
B	number of bootstrap replicates to be generated
J	number of observations used to build the bands for the MBD computation. Currently, only the value J=2 can be used
...	additional arguments to be passed to the kmeans function for the final clustering; at this stage nstart is set to 1, as the initial seeds are fixed

Details

The brik algorithm is a simple, computationally feasible method, which provides k-means with a set of initial seeds to cluster datasets of arbitrary dimensions. It consists of two stages: first, a set of cluster centers is obtained by applying k-means to bootstrap replications of the original data to be, next, clustered; the deepest point in each assembled cluster is returned as initial seeds for k-means.

Value

seeds	a matrix of size k x d containing the initial seeds obtained with the BRIk algorithm
km	an object of class kmeans corresponding to the run of kmeans on x with starting points seeds

Author(s)

Javier Albert Smet <javas@kth.se> and Aurora Torrente <etorrent@est-econ.uc3m.es>

References

Torrente, A. and Romo, J. (2020). Initializing k-means Clustering by Bootstrap and Data Depth. *J Classif* (2020). https://doi.org/10.1007/s00357-020-09372-3.

Examples

```
## brik algorithm
    ## simulated data
    set.seed(0)
    g1 <- matrix(rnorm(200,0,3), 25, 8) ; g1[,1]<-g1[,1]+4;
    g2 <- matrix(rnorm(200,0,3), 25, 8) ; g2[,1]<-g2[,1]+4; g2[,3]<-g2[,3]-4
    g3 <- matrix(rnorm(200,0,3), 25, 8) ; g3[,1]<-g3[,1]+4; g3[,3]<-g3[,3]+4

    x <- rbind(g1,g2,g3)
    labels <-c(rep(1,25),rep(2,25),rep(3,25))

    C1 <- kmeans(x,3)
    C2 <- brik(x,3,B=25)

    table(C1$cluster, labels)
    table(C2$km$cluster, labels)
```

---

elbowRule — *Selection of Appropriate DF Parameter Based on an Elbow Rule for the Distortion*

Description

`elbowRule` runs the FABRIk algorithm for different degrees of freedom (DF) and suggests the best of such values as the one where the minimum distortion is obtained. An optional visualization of the computed values allows the choice of alternative suitable DF values based on an elbow-like rule.

Usage

```
elbowRule(x, k, method="Ward", nstart=1, B = 10, J = 2, x.coord = NULL, OSF = 1,
    vect = NULL, intercept = TRUE, degPolyn = 3, degFr = 4:20, knots = NULL,
    plot = FALSE, ...)
```

Arguments

`x`	a data matrix containing `N` observations (individuals) by rows and `d` variables (features) by columns
`k`	number of clusters
`method`	clustering algorithm used to cluster the cluster centres from the bootstrapped replicates; `Ward`, by default. Currently, only `pam` and randomly initialised `kmeans` are implemented

nstart	number of random initialisations when using the kmeans method to cluster the cluster centres
B	number of bootstrap replicates to be generated
J	number of observations used to build the bands for the MBD computation. Currently, only the value J=2 can be used
x.coord	initial x coordinates (time points) where the functional data is observed; if not provided, it is assumed to be 1:d
OSF	oversampling factor for the smoothed data; an OSF of m means that the number of (equally spaced) time points observed in the approximated function is m times the number of original number of features, d
vect	optional collection of x coordinates (time points) where to assess the smoothed data; if provided, it ignores the OSF
intercept	if TRUE, an intercept is included in the basis; default is FALSE
degPolyn	degree of the piecewise polynomial; 3 by default (cubic splines)
degFr	a vector containing tentative values of the degrees of freedom, to be tested
knots	the internal breakpoints that define the spline
plot	a Boolean parameter; it allows plotting the distortion against the degrees of freedom. Set to FALSE by default
...	additional arguments to be passed to the kmeans function for the final clustering; at this stage nstart is set to 1, as the initial seeds are fixed

Details

The function implements a simple elbow-like rule that allows selecting an appropriate value for the DF parameter among the tested ones. It computes the distortion obtained for each of these values and returns the one yielding to the smallest distortion. By setting the parameter plot to TRUE the distortion is plotted against the degrees of freedom and elbows or minima can be visually detected.

Value

df	the original vector of DF values to be tested
tot.withinss	a vector containing the distortion obtained for each tested DF value
optimal	DF value producing the smallest distortion among the tested df

Author(s)

Javier Albert Smet <javas@kth.se> and Aurora Torrente <etorrent@est-econ.uc3m.es>

References

Torrente, A. and Romo, J. (2020). Initializing Kmeans Clustering by Bootstrap and Data Depth. *J Classif* (2020). https://doi.org/10.1007/s00357-020-09372-3. Albert-Smet, J., Torrente, A. and Romo J. (2021). Modified Band Depth Based Initialization of Kmeans for Functional Data Clustering. Submitted to Computational Statistics and Data Analysis.

Examples

```
## simulated data
set.seed(1)
x.coord = seq(0,1,0.01)
x <- matrix(ncol = length(x.coord), nrow = 100)
labels <- matrix(ncol = 100, nrow = 1)

centers <-  matrix(ncol = length(x.coord), nrow = 4)
centers[1, ] <- abs(x.coord)-0.5
centers[2, ] <- (abs(x.coord-0.5))^2 - 0.8
centers[3, ] <- -(abs(x.coord-0.5))^2 + 0.7
centers[4, ] <- 0.75*sin(8*pi*abs(x.coord))

for(i in 1:4){
    for(j in 1:25){
        labels[25*(i-1) + j] <- i
        if(i == 1){x[25*(i-1) + j, ] <- abs(x.coord)-0.5 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 2){x[25*(i-1) + j, ] <- (abs(x.coord-0.5))^2 - 0.8 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 3){x[25*(i-1) + j, ] <- -(abs(x.coord-0.5))^2 + 0.7 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 4){x[25*(i-1) + j, ] <- 0.75*sin(8*pi*abs(x.coord)) +
            rnorm(length(x.coord),0,1.5)}
        }
    }

ER <- elbowRule(x, 4, B=25, degFr = 5:12, plot=FALSE)
ER <- elbowRule(x, 4, B=25, degFr = 5:12, plot=TRUE)
```

`fabrik` *Computation of Initial Seeds for Kmeans and Clustering of Functional Data*

Description

`fabrik` fits splines to the multivariate dataset and runs the BRIk algorithm on the smoothed data. For functional data, this is just a straight forward application of BRIk to the k-means algorithm; for multivariate data, the result corresponds to an alternative clustering method where the objective function is not necessarily minimised, but better allocations are obtained in general.

Usage

```
fabrik(x, k, method="Ward", nstart=1, B = 10, J = 2, x.coord = NULL, OSF = 1,
    vect = NULL, intercept = TRUE, degPolyn = 3, degFr = 5, knots = NULL, ...)
```

Arguments

x	a data matrix containing N observations (individuals) by rows and d variables (features) by columns
k	number of clusters
method	clustering algorithm used to cluster the cluster centres from the bootstrapped replicates; Ward, by default. Currently, only pam and randomly initialised kmeans are implemented
nstart	number of random initialisations when using the kmeans method to cluster the cluster centres
B	number of bootstrap replicates to be generated
J	number of observations used to build the bands for the MBD computation. Currently, only the value J=2 can be used
x.coord	initial x coordinates (time points) where the functional data is observed; if not provided, it is assumed to be 1:d
OSF	oversampling factor for the smoothed data; an OSF of m means that the number of (equally spaced) time points observed in the approximated function is m times the number of original number of features, d
vect	optional collection of x coordinates (time points) where to assess the smoothed data; if provided, it ignores the OSF
intercept	if TRUE, an intercept is included in the basis; default is FALSE
degPolyn	degree of the piecewise polynomial; 3 by default (cubic splines)
degFr	degrees of freedom, as in the bs function
knots	the internal breakpoints that define the spline
...	additional arguments to be passed to the kmeans function for the final clustering; at this stage nstart is set to 1, as the initial seeds are fixed

Details

The FABRIk algorithm extends the BRIk algorithm to the case of longitudinal functional data by adding a step that includes B-splines fitting and evaluation of the curve at specific x coordinates. Thus, it allows handling issues such as noisy or missing data. It identifies smoothed initial seeds that are used as starting points of kmeans on the smoothed data. The resulting clustering does not optimise the distortion (sum of squared distances of each data point to its nearest centre) in the original data space but it provides in general a better allocation of datapoints to real groups.

Value

seeds	a matrix of size k x D, where D is either m x d or the length of vect . It contains the initial smoothed seeds obtained with the BRIk algorithm
km	an object of class kmeans corresponding to the run of kmeans on the smoothed data, with starting points seeds

Author(s)

Javier Albert Smet <javas@kth.se> and Aurora Torrente <etorrent@est-econ.uc3m.es>

References

Torrente, A. and Romo, J. (2020). Initializing Kmeans Clustering by Bootstrap and Data Depth. *J Classif* (2020). https://doi.org/10.1007/s00357-020-09372-3. Albert-Smet, J., Torrente, A. and Romo J. (2021). Modified Band Depth Based Initialization of Kmeans for Functional Data Clustering. Submitted to Computational Statistics and Data Analysis.

Examples

```
## fabrik algorithm
    ## simulated data
    set.seed(1)
    x.coord = seq(0,1,0.01)
    x <- matrix(ncol = length(x.coord), nrow = 100)
    labels <- matrix(ncol = 100, nrow = 1)

    centers <-  matrix(ncol = length(x.coord), nrow = 4)
    centers[1, ] <- abs(x.coord)-0.5
    centers[2, ] <- (abs(x.coord-0.5))^2 - 0.8
    centers[3, ] <- -(abs(x.coord-0.5))^2 + 0.7
    centers[4, ] <- 0.75*sin(8*pi*abs(x.coord))

    for(i in 1:4){
        for(j in 1:25){
            labels[25*(i-1) + j] <- i
            if(i == 1){x[25*(i-1) + j, ] <- abs(x.coord)-0.5 +
                rnorm(length(x.coord),0,1.5)}
            if(i == 2){x[25*(i-1) + j, ] <- (abs(x.coord-0.5))^2 - 0.8 +
                rnorm(length(x.coord),0,1.5)}
            if(i == 3){x[25*(i-1) + j, ] <- -(abs(x.coord-0.5))^2 + 0.7 +
                rnorm(length(x.coord),0,1.5)}
            if(i == 4){x[25*(i-1) + j, ] <- 0.75*sin(8*pi*abs(x.coord)) +
                rnorm(length(x.coord),0,1.5)}
            }
        }

    C1 <- kmeans(x,4)
    C2 <- fabrik(x,4,B=25)

    table(C1$cluster, labels)
    table(C2$km$cluster, labels)
```

---

plotKmeansClustering *Kmeans Clustering Plot*

Description

plotKmeansClustering represents, in different subpanels, each of the clusters obtained after running k-means. The corresponding centroid is highlighted.

Usage

```
plotKmeansClustering(x, kmeansObj, col=c(8,2), lty=c(2,1), x.coord = NULL,
    no.ticks = 5, ...)
```

Arguments

x	a data matrix containing N observations (individuals) by rows and d variables (features) by columns
kmeansObj	an object of class kmeans, containing the cluster labels output by kmeans
col	a vector containing colors for the elements in x and for the centroid. The last one is used for the centroid, whereas the previous ones are recycled
lty	a vector containing the line type for the elements in x and for the centroid. The last one is used for the centroid, whereas the previous ones are recycled
x.coord	initial x coordinates (time points) where the functional data is observed; if not provided, it is assumed to be 1:d
no.ticks	number of ticks to be displayed in the X axis
...	additional arguments to be passed to the plot function

Details

The function creates a suitable grid where to plot the different clusters independently. In the i-th cell of the grid, the data points corresponding to the i-th cluster are represented in parallel coordinates and the final centroid is highlighted.

Value

the function returns invisibly a list with the following components:

clusters	a list containing one cluster per component; observations are given by rows
centroids	a list with the centroid of each cluster

Author(s)

Javier Albert Smet <javas@kth.se> and Aurora Torrente <etorrent@est-econ.uc3m.es>

Examples

```
## simulated data
set.seed(1)
x.coord = seq(0,1,0.01)
x <- matrix(ncol = length(x.coord), nrow = 100)
labels <- matrix(ncol = 100, nrow = 1)

centers <-  matrix(ncol = length(x.coord), nrow = 4)
centers[1, ] <- abs(x.coord)-0.5
centers[2, ] <- (abs(x.coord-0.5))^2 - 0.8
centers[3, ] <- -(abs(x.coord-0.5))^2 + 0.7
centers[4, ] <- 0.75*sin(8*pi*abs(x.coord))
```

```
for(i in 1:4){
    for(j in 1:25){
        labels[25*(i-1) + j] <- i
        if(i == 1){x[25*(i-1) + j, ] <- abs(x.coord)-0.5 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 2){x[25*(i-1) + j, ] <- (abs(x.coord-0.5))^2 - 0.8 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 3){x[25*(i-1) + j, ] <- -(abs(x.coord-0.5))^2 + 0.7 +
            rnorm(length(x.coord),0,1.5)}
        if(i == 4){x[25*(i-1) + j, ] <- 0.75*sin(8*pi*abs(x.coord)) +
            rnorm(length(x.coord),0,1.5)}
        }
    }

plotKmeansClustering(x, kmeans(x,4))
plotKmeansClustering(x, brik(x,4)$km)
plotKmeansClustering(x, fabrik(x,4)$km)
plotKmeansClustering(x, fabrik(x,4,degFr=10)$km)
```

Index