Package 'kmed'

January 2, 2019

Type Package

Title Distance-Based k-Medoids

Version 0.2.0

Date 2019-01-02

Author Weksi Budiaji

Maintainer Weksi Budiaji <budiaji@untirta.ac.id>

Description Algorithms of distance-based k-medoids clustering:
simple and fast k-medoids (Park and Jun, 2009) <doi:10.1016/j.eswa.2008.01.039>,
ranked k-medoids (Zadegan et al., 2013) <doi:10.1016/j.knosys.2012.10.012>, and
step k-medoids (Yu et al., 2018) <doi:10.1016/j.eswa.2017.09.052>.
Calculate distances for mixed variable data such as Gower, Podani, Wishart (2003) <doi:10.1007/978-3-642-55721-7_23>,
Huang (1997) <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.9984&rep=rep1&type=pdf>,
Harikumar and PV (2015) <doi:10.1016/j.procs.2015.10.077>, and
Ahmad and Dey (2007) <doi:10.1016/j.datak.2007.03.016>.
Cluster validation applies bootstrap procedure producing a heatmap with a flexible reordering matrix algorithm such as complete, ward, or average linkages.

Depends R (>= 2.10)

License GPL-3

LazyData TRUE

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports ggplot2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-02 18:30:02 UTC

R topics documented:

`barplotnum` *Barplot of each cluster for numerical variables data set.*

Description

This function create a barplot from a cluster result.

Usage

```
barplotnum(dataori, clust, nc = 1)
```

Arguments

`dataori`	An original data set.
`clust`	A vector of cluster membership.
`nc`	A number of column of the plot.

Details

This is a function to produce a barplot for each cluster.

Value

Function returns a barplot.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
dat <- iris[,1:4]
memb <- cutree(hclust(dist(dat)),3)
barplotnum(dat, memb)
barplotnum(dat, memb, 2)
```

---

clust4 *4 clustered data*

Description

A dataset containing two variables of 300 objects and their class memberships generated by a clusterGeneration package.

Usage

```
clust4
```

Format

A data frame with 300 rows and 3 variables:

x1 X1.

x2 X2.

class Class membership.

---

clustboot *Bootstrap replications for clustering alorithm*

Description

This function do bootstrap replications for a cluster algorithm.

Usage

```
clustboot(distdata, nclust = 2, algorithm, nboot = 25, diss = TRUE)
```

Arguments

distdata	A matrix of distance objects (n x n) or a diss class.
nclust	The number of clusters
algorithm	Any clustering algorithm function with the end result being only membership.
nboot	The number of bootstrap replicates
diss	A logical if the distdata is a dist or matrix object.

Details

This is a function to obtain bootstrap evaluation for a cluster. The cluster matrix can be further analyzed. In the algorithm function, the input arguments are only a distance/ matrix and a number of cluster.The output is only the membership.

Value

Function returns a bootstrap cluster matrix (n x number of bootstrap replicates).

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
parkboot <- function(x, nclust) {
res <- fastkmed(x, nclust, iterate = 50)
return(res$cluster)
}
irisboot <- clustboot(mrwdist, nclust=3, parkboot, nboot=7)
head(irisboot)
```

---

clustheatmap *Consensus matrix heatmap from A consensus matrix*

Description

This function create a consensus matrix heatmap from a consensus matrix.

Usage

```
clustheatmap(consmat, title = "")
```

Arguments

consmat	A matrix of consensus matrix.
title	A character of plot title

Details

This is a function to produce a consensus matrix heatmap from a consensus matrix.

Value

Function returns a heatmap plot.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
parkboot <- function(x, nclust) {
res <- fastkmed(x, nclust, iterate = 50)
return(res$cluster)
}
irisboot <- clustboot(mrwdist, nclust=3, parkboot, nboot=7)
wardorder <- function(x, nclust) {
res <- hclust(x, method = "ward.D2")
member <- cutree(res, nclust)
return(member)
}
consensusiris <- consensusmatrix(irisboot, nclust = 3, wardorder)
clustheatmap(consensusiris)
```

---

consensusmatrix *Consensus matrix from A bootstrap replicate matrix*

Description

This function create a consensus matrix from a bootstrap replicate matrix.

Usage

```
consensusmatrix(bootdata, nclust, reorder)
```

Arguments

bootdata	A matrix of bootstrap replicate (n x b) membership.
nclust	The number of clusters
reorder	Any clustering algorithm function with the input is a distance and the end result being only membership.

Details

This is a function to obtain a consensus matrix from a bootstrap evaluation for a cluster. The consensus matrix can be further plotted.

Value

Function returns a consensus matrix (n x n).

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
parkboot <- function(x, nclust) {
res <- fastkmed(x, nclust, iterate = 50)
return(res$cluster)
}
irisboot <- clustboot(mrwdist, nclust=3, parkboot, nboot=7)
wardorder <- function(x, nclust) {
res <- hclust(x, method = "ward.D2")
member <- cutree(res, nclust)
return(member)
}
consensusiris <- consensusmatrix(irisboot, nclust = 3, wardorder)
consensusiris[c(1:5,51:55,101:105),c(1:5,51:55,101:105)]
```

---

cooccur *Co-occurrence distance for binary/ categorical variables data.*

Description

This function computes and returns the distance matrix computed by co-occurence distance.

Usage

```
cooccur(data)
```

Arguments

data	A matrix or data frame of binary/ categorical variables. The values of matrix should be integer, i.e 1, 2, 3, ..., or will be converted to integer otherwise.

Details

This is a function to compute a co-occurence distance. It returns a matrix of distance objects , i.e n x n.

Value

A matrix of distance from binary/ categorical variable.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

Harikumar, S., PV, S., 2015. K-medoid clustering for heterogeneous data sets. JProcedia Computer Science 70, 226-237.

Examples

```
set.seed(1)
a <- matrix(sample(1:2, 7*3, replace = TRUE), 7, 3)
cooccur(a)
```

---

distmix *Distances for mixed variables.*

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the mixed variable data.

Usage

```
distmix(data, method = "gower", idnum = NULL, idbin = NULL,
  idcat = NULL)
```

Arguments

data	A data frame or a matrix object.
method	A distance for mixed variables: "gower", "wishart", "podani", "huang", "harikumar", and "ahmad".
idnum	A vector of index of numerical variables.
idbin	A vector of index of binary variables.
idcat	A vector of index of categorical variables.

Details

This is a function to compute distance of mixed variable data. It returns a matrix of all object distances. The available distance are Gower ("gower"), Wishart ("wishart"), Podani ("podani"), Huang ("huang"), Harikumar-PV ("harikumar"), Achmad-Dey ("ahmad"). Because it computes distance of mixed variable data, at least two different class of variables in idnum, idbin, or idcat must be supplied, such as numerical and binary or binary and categorical indices.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

Ahmad, A., and Dey, L. 2007. A K-mean clustering algorithm for mixed numeric and categorical data. Data and Knowledge Engineering 63, 503-527.

Gower, J., 1971. A general coefficient of similarity and some of its properties. Biometrics 27, 857-871

Harikumar, S., PV, S., 2015. K-medoid clustering for heterogeneous data sets. JProcedia Computer Science 70, 226-237.

Huang, Z., 1997. Clustering large data sets with mixed numeric and categorical values, in: The First Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 21-34.

Podani, J., 1999. Extending gower's general coefficient of similarity to ordinal characters. Taxon 48, 331-340.

Wishart, D., 2003. K-means clustering with outlier detection, mixed variables and missing values, in: Exploratory Data Analysis in Empirical Research: Proceedings of the 25th Annual Conference of the Gesellschaft fur Klassifikation e.V., University of Munich, March 14-16, 2001, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 216-226.

Examples

```
set.seed(1)
a <- matrix(sample(1:2, 7*3, replace = TRUE), 7, 3)
a1 <- matrix(sample(1:3, 7*3, replace = TRUE), 7, 3)
mixdata <- cbind(iris[1:7,1:3], a, a1)
colnames(mixdata) <- c(paste(c("num"), 1:3, sep = ""),
                       paste(c("bin"), 1:3, sep = ""),
```

```
                        paste(c("cat"), 1:3, sep = ""))
distmix(mixdata, method = "gower", idnum = 1:3, idbin = 4:6, idcat = 7:9)
```

`distNumeric` *A pair distance for continuous variables.*

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the pairwise distances between the rows of two data of numerical variables.

Usage

```
distNumeric(x, y, method = "mrw")
```

Arguments

`x`	A data matrix.
`y`	A second data matrix.
`method`	A distance for numerical variables.

Details

This is a two-data-set to compute distance. It returns a matrix of all pairwise distances between rows in x and y. The available distance are Manhattan weighted by rank ("mrw"), Squared Euclidean weighted by variance ("sev"), Squared Euclidean weighted by rank ("ser"), and Squared Euclidean ("se").

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
mrwdist[1:6,1:6]
```

`fastkmed` *Simple and fast k-medoid algorithm from Park and Jun.*

Description

This function computes and returns the clustering result computed by using a specified distance via Park and Jun's algorithm.

Usage

```
fastkmed(distdata, ncluster, iterate = 10, init = NULL)
```

Arguments

`distdata`	A matrix of distance objects (n x n) or a diss class.
`ncluster`	A number of cluster.
`iterate`	A number of iteration for clustering algorithm.
`init`	An index of the initial medoids.

Details

This is a k-medoids algorithm that has been proposed by Park and Jun. The algorihm has been claimed to be fast and simple. The medoids updating in this algorihm is similar to kmeans centroid updating.

Value

Function returns a partitioning clustering algorithm result consists of cluster membership, cluster medoid, the minimum distance to the cluster medoid.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

Park, H., Jun, C., 2009. A simple and fast algorithm for k-medoids clustering. Expert Systems with Applications 36, 3336-3341.

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
result <- fastkmed(mrwdist, ncluster = 3, iterate = 50)
table(result$cluster, iris[,5])
```

globalfood *Global food security index*

Description

A dataset containing four variables of 113 countries for their food security index based on panelists evaluation.

Usage

```
globalfood
```

Format

A data frame with 113 rows and 4 variables:

affordability affordability

availability availability

safety safety

resilience resilience

Source

http://foodsecurityindex.eiu.com

matching *A pair distance for binary/ categorical variables.*

Description

This function computes and returns the distance matrix computed by using the simple matrching distance.

Usage

```
matching(x, y)
```

Arguments

x	A data frame/ matrix.
y	A second data frame/ matrix.

Details

This is a function to compute simple matching distance. It returns a matrix of distance objects, i.e n x n.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
set.seed(1)
a <- matrix(sample(1:2, 7*3, replace = TRUE), 7, 3)
matching(a, a)
```

---

pcabiplot *Biplot of a PCA object*

Description

This function create a biplot from a pca object generated by prcomp function.

Usage

```
pcabiplot(PC, x = "PC1", y = "PC2", var.line = TRUE, colobj = rep(1,
  nrow(PC$x)))
```

Arguments

PC	A pca object generated by prcomp function.
x	X axis.
y	Y axis.
var.line	A logical input, if variable lines are plotted.
colobj	A vector for giving colours to the objects.

Details

This is a function to produce a pca biplot from prcomp function.

Value

Function returns a pca biplot.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

Examples

```
pcadat <- prcomp(iris[,1:4], scale. = TRUE)
pcabiplot(pcadat)
```

rankkmed *Rank k-medoid algorithm from Zadegan et. al.*

Description

This function computes and returns the clustering result computed by using a specified distance via rank k-medoids algorithm.

Usage

```
rankkmed(distdata, ncluster, m = 3, iterate = 10, initial = NULL)
```

Arguments

distdata	A matrix of distance objects (n x n) or a diss class.
ncluster	A number of cluster.
m	A number of objects to compute hostility.
iterate	A number of iteration for clustering algorithm.
initial	A vector of initial objects as the cluster medoids.

Details

This is a k-medoids algorithm that has been proposed by Zadegan et. al. The algorihm has been claimed to be suitable for large dataset. The medoids updating in this algorihm is similar to kmeans centroid updating.

Value

Function returns a partitioning clustering algorithm result consists of cluster membership, cluster medoid, the minimum distance to the cluster medoid.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

Zadegan, S.M.R, Mirzaie M, and Sadoughi, F. 2013. Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. Knowledge-Based Systems 39, 133-143.

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
result <- fastkmed(mrwdist, ncluster = 3, iterate = 50)
table(result$cluster, iris[,5])
```

---

shadow *Centroid shadow value (CSV) index of each cluster based on medoid.*

Description

This function create a centroid shadow value index using medoid (instead of centroid) and its plot.

Usage

```
shadow(distdata, idmedoid, idcluster)
```

Arguments

distdata	A distance object/ a n x n distance matrix.
idmedoid	A vector of id medoids.
idcluster	A vector of cluster membership.

Details

This is a function to produce a centroid shadow value index (using medoid) and its plot for each cluster. The id medoids must match with the cluster membership, for example, if the id medoids are 2, 25, and 57 (3 medoids), the idcluster must have 3 unique memberships.

Value

Function returns a shadow value index and plot.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

F. Leisch. 2010 Neighborhood graphs, stripes and shadow plots for cluster visualization. Statistics and Computing. vol. 20, pp. 457-469

Examples

```
distiris <- as.matrix(dist(iris[,1:4]))
res <- fastkmed(distiris, 3)
sha <- shadow(distiris, res$medoid, res$cluster)
sha$result[c(1:3,70:75,101:103),]
sha$plot
```

`silhoutte` *Silhoutte index of each cluster.*

Description

This function create a silhoutte index and its plot.

Usage

```
silhoutte(distdata, idmedoid, idcluster)
```

Arguments

`distdata`	A distance object/ a n x n distance matrix.
`idmedoid`	A vector of id medoids.
`idcluster`	A vector of cluster membership.

Details

This is a function to produce a silhoutte index and its plot for each cluster. The id medoids must match with the cluster membership, for example, if the id medoids are 2, 25, and 57 (3 medoids), the idcluster must have 3 unique memberships.

Value

Function returns a silhoutte index and plot.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

P. J. Rousseeuw. 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, vol. 20, pp. 53-65

Examples

```
distiris <- as.matrix(dist(iris[,1:4]))
res <- fastkmed(distiris, 3)
silh <- silhoutte(distiris, res$medoid, res$cluster)
silh$result[c(1:3,70:75,101:103),]
silh$plot
```

`stepkmed` *Step k-medoid algorithm from Yu et al.*

Description

This function computes and returns the clustering result computed by using a specified distance via Yu et al. algorithm.

Usage

```
stepkmed(distdata, ncluster, iterate = 10, alpha = 1)
```

Arguments

`distdata`	A matrix of distance objects (n x n) or a diss class.
`ncluster`	A number of cluster.
`iterate`	A number of iteration for clustering algorithm.
`alpha`	A numeric number to determine the range of initial medoids selection.

Details

This is a k-medoids algorithm that has been proposed by Yu et al. The algorihm has been claimed to be a remedy of simple and fast k-medoid. The medoids updating in this algorihm is similar to simple and fast k-medoid.

Value

Function returns a partitioning clustering algorithm result consists of cluster membership, cluster medoid, the minimum distance to the cluster medoid.

Author(s)

Weksi Budiaji
Contact: <budiaji@untirta.ac.id>

References

Yu, D., Liu, G., Guo, M., Liu, X., 2018. An improved K-medoids algorithm based on step increasing and optimizing medoids. Expert Systems with Applications 92, 464-473.

Examples

```
num <- as.matrix(iris[,1:4])
mrwdist <- distNumeric(num, num, method = "mrw")
result <- stepkmed(mrwdist, ncluster = 3, iterate = 50, alpha = 1.5)
table(result$cluster, iris[,5])
```

Index