Package 'clustTool'

January 2, 2012

Type Package

Title GUI for clustering data with spatial information

Version 1.6.5

Date 2010-08-16

Author Matthias Templ

Maintainer Matthias Templ <templ@statistik.tuwien.ac.at>

Depends R (>= 2.2.0), mvoutlier, cluster, randomForest, tcltk, e1071,rpart, class, grid, mclust, flexclust, vegan, clue, mvpart,lattice, kernlab, compositions

Description This package can be used for clustering data with spatial information. Try function GUIspatClust() to run the clustTool GUI.

License GPL (>= 2)

Repository CRAN

Date/Publication 2010-08-16 16:12:10

R topics documented:

clustTool-package *Clustering with spatial information.*

Description

Cluster results can change dramatically depending on the choice of the clustering method, the distance measure, and the number of clusters. Moreover, depending on the selected validity measure, there may be different results for the optimal number of clusters. Despite of the changing cluster results, each partition could still be informative and valuable. The results can give an interesting insight into the multivariate data structure even if the validity measure does not suggest the optimum for the chosen cluster number. It is thus desirable to perform cluster analysis in an exploratory context, by changing the cluster parameters and inspecting the results visually.

For this purpose, this statistical tool has been developed. Data, subsets of the data, coordinates and maps can be selected. Furhtermore, different parameters like the distance measure, the clustering method, the number of clusters and the validity measure can be selected. Depending on the selection the clusters can be presented on maps. Additionally, plots of the cluster centres are provided.

Details

Package:	clustTool
Type:	Package
Version:	1.0
Date:	2006-08-09
License:	GPL (>= 2)

Try

GUIspatStat

Author(s)

Matthias Templ

Maintainer: Matthias Templ <templ@statistik.tuwien.ac.at>

Examples

```
## destroy the Rcmdr Commander window
## tkdestroy(commanderWindow)
library(mvoutlier)
data(kola.background)
data(humus)
x <- prepare(humus[,c("As", "Ca", "Co", "Mo", "Ni")])
cl1 <- clust(x, k=9, method="clara", distMethod="manhattan")
names(cl1)
clustPlot(coord=humus[,2:3], clust=cl1, k=cl1$k, val="median.distance",
          Map="kola.background")
```

```
GUIspatClust()
```

---

clust *Wrapper function for a variety of clustering algorithms*

Description

Performs cluster analysis on data.

Usage

```
clust(x = Cassini$x, k = 3, method = "kmeansHartigan", seed = set.seed(123),
      distMethod = "euclidean", qtclustsize = 0.7, iter.max = 100, eps = 0.1,
      vals = TRUE, alt = NULL, coord = NULL, bic = NULL)
```

Arguments

x	data frame or matrix
k	Number of clusters
method	Cluster algorithm
seed	Seed (can be useful if results from clustering should be reproduced exactly)
distMethod	Distance Measure
qtclustsize	Only important if method qtclust is chosen (see 'qtclust' in package **flexclust**)
iter.max	Only important if method kmeans is chosen (see 'kmeans' in package **stats**
eps	Only important if method 'dbscan' is chosen
vals	Validity measures for the resulting clusters would be calculated if this parameter is set to TRUE
alt	an integer vector for each observation indicating the cluster number for an alternative clustering. If provided, the corrected rand index for 'clustering' vs. 'alt.clustering' will be computed (see also in package **fpc**).
coord	Cluster validity measures will be calculated based on coordinates.
bic	Alternative way to specify bic values for each cluster.

Details

This function acts like a wrapper function for applying a variety of clustering algorithms. The function would be carried out from the **clustTool**-GUI. To specify additional parameters for special algorithms one should use the algorithm itself and structure the output as the output from this function (as class 'clust' suggests).

Number of Clusters: Since there will be no necessarity for a large number of clusters, the maximum number of clusters should not exeed 12.

Cluster algorithms: Possible values are: "kmeansHartigan", "kmeansLloyd", "kmeansForgy", "kmeansMacQueen", "cmeans", "cmeansUfcl", "pam", "clara", "fanny", "bclust", "cshell", "Mclust", "kccaKmeans", "kccaKmedians", "kccaAngle", "kccaJaccard", "kccaEjaccard", "cclustKmeans", "cclustHardcl",

"cclustNeuralgas", "qtclustKmeans", "qtclustKmedian", "qtclustAngle", "qtclustJaccard", "qtclustEjaccard", "dbscan", "speccPolydot", "fixmahal", "hclustSingle", "hclustComplete", "hclustAverage", "hclustWard", "hclustMcquitty", "hclustMedian", "hclustcentroid".

Cluster algorithms which are supported by **clustTool**-GUI: "kmeansHartigan", "clara", "bclust", "Mclust", "kccaKmeans", "speccPolydot", "cclustNeuralgas", "cmeans", "kccaKmedians".

For details see the help files listed below.

distMethod: Possible values are: "euclidean", "manhattan", "maximum", "canberra", "cosa", "rf" (dissimilarity measure based on random Forest proximity measure), "gower", "bray", "kulczynski", "chord", "morisita", "horn", "mountford", "correlation" (dissimilarity measure based on correlations).

Distance measures which are supported by **spatClust**-GUI: "euclidean", "manhattan","rf","bray","gower","kulczynski", "morisita","correlation".

For details see the help files listed below.

Value

`cluster`	A vector of integers indicating the cluster to which each point is allocated.
`centers`	A matrix of cluster centres.
`size`	The number of points in each cluster.
`xdata`	The input data.
`method`	Clustering method
`distMethod`	Distance measure
`k`	Number of clusters
`valTF`	logical, if global validity measures provided
`valMeasures`	global validity measures
`silwidths`	local validity measure
`separation`	local validity measure
`diameter`	local validity measure
`average.distance`	local validity measure
`median.distance`	local validity measure
`average.toother`	local validity measure
`vp`	logical, if colnames provided

Author(s)

Matthias Templ

See Also

Clustering methods:

kmeans, cmeans, pam, clara, fanny, bclust, Mclust, kcca, cclust, specc, hclust

Distance measures:

dist, vegdist, randomForest, "cosa", cor

Cluster validity measures:

cluster.stats

Examples

```
require(mvoutlier)
data(humus)
x <- prepare(humus[,c("As", "Ca", "Co", "Mo", "Ni")])
cl1 <- clust(x, k=9, method="clara", distMethod="manhattan")
cl1
names(cl1)
```

`clustPlot` *Plotting function for package clustTool*

Description

The first plot visualises the clusters in a map. The second plot provides the visualisation of the cluster centres centres.

Usage

```
clustPlot(coord, clust, k, val = "silwidths", which.plot = c(1, 2), Map = "kola.background", texth = 0.7
```

Arguments

`coord`	(x,y)-coordinates of the data
`clust`	Object from function "clust"
`k`	Number of clusters
`val`	local validity measure
`which.plot`	if a subset of the plots is required, specify a subset of the numbers '1:2'
`Map`	A map may be selected
`texth`	cex of variable names in the cluster centres plot.

Details

The resulting clusters of function "clust" will be visualised in maps.

In general, not only the location of the single clustes in the map is of interest but also the composition of the single cluster. For this purpose a plot of the cluster centres is supported which is helpful for the interpretation of the clusters. The cluster centre is the element-wise mean of all observations of a cluster. Therefore, for each cluster all elements used for clustering are presented. The resulting means for all clusters are horizontally arranged. If the variables used for clustering were standardised they have the same contribution for the cluster analysis. If single elements show very high or low means for a cluster they are highly influential for this cluster.

Author(s)

Matthias Templ

See Also

clust, GUIspatClust

Examples

```
library(mvoutlier)
data(kola.background)
data(humus)
x <- prepare(humus[,c("As", "Ca", "Co", "Mo", "Ni")])
cl1 <- clust(x, k=9, method="clara", distMethod="manhattan")
names(cl1)
clustPlot(coord=humus[,2:3], clust=cl1, k=cl1$k, val="median.distance",
          Map="kola.background")
```

GUIspatClust *GUI for package clustTool*

Description

Graphical user interface (GUI) for package **clustTool**.

Usage

```
GUIspatClust()
```

Details

Cluster results can change dramatically with the choice of the clustering method, the distance measure, and the number of clusters. Moreover, depending on the selected validity measure, a different answers result for the optimal number of clusters. Despite of the changing cluster results, each partition could still be informative and valuable. The results can give an interesting insight into the multivariate data structure even if the validity measure does not suggest the optimum for the

chosen cluster number. It is thus desirable to perform cluster analysis in an exploratory context, by changing the cluster parameters and visually inspecting the results.

For this purpose, this statistical tool has been developed. Data, subsets of the data, coordinates and maps can be selected. Different parameters like the distance measure, the clustering method, the number of clusters, and the validity measure can be chosen. Depending on the selection, the clusters can be presented in maps and plots of the cluster centres are provided.

Value

The plots of interest.

Additionally, informations about the cluster result are printed in the R console.

Author(s)

Matthias Templ

See Also

clust, clustPlot,prepare

Examples

```
GUIspatClust()
```

`kola.background` *kola background*

Description

Coordinates of the kolamap background

Usage

```
data(kola.background)
```

Format

The format is: List of 4 $ boundary:'data.frame': 50 obs. of 2 variables: ..$ V1: num [1:50] 388650 388160 386587 384035 383029 ... ..$ V2: num [1:50] 7892400 7881248 7847303 7790797 7769214 ... $ coast :'data.frame': 6259 obs. of 2 variables: ..$ V1: num [1:6259] 438431 439102 439102 439643 439643 ... ..$ V2: num [1:6259] 7895619 7896495 7896495 7895800 7895542 ... $ borders :'data.frame': 504 obs. of 2 variables: ..$ V1: num [1:504] 417575 417704 418890 420308 422731 ... ..$ V2: num [1:504] 7612984 7612984 7613293 7614530 7615972 ... $ lakes :'data.frame': 6003 obs. of 2 variables: ..$ V1: num [1:6003] 547972 546915 NA 547972 547172 ... ..$ V2: num [1:6003] 7815109 7815599 NA 7815109 7813873 ...

Details

Is used by map.plot() in package **mvoutlier** and used by function 'clustPlot'.

Source

Kola Project (1993-1998)

References

see reference in **mvoutlier**.

Examples

```
library(mvoutlier)
data(kola.background)
   example(map.plot)
```

`plotKola.background` *plot kola.background*

Description

Function for visualising 'kola.background'.

Usage

```
plotKola.background(map = "kola.background", which.map = c(1, 2, 3, 4),
           map.col = c(5, 1, 3, 4), map.lwd = c(2, 1, 2, 1), add.plot = FALSE)
```

Arguments

`map`	default: 'kola.background'
`which.map`	1: plot project boundary, 2: plot coast line, 3: plot country borders, 4: plot lakes and rivers
`map.col`	Line colors for project boundary, coast line, country borders, lakes and rivers
`map.lwd`	Line widths for project boundary, coast line, country borders, lakes and rivers
`add.plot`	if FALSE (default) create new plot

Author(s)

Peter Filzmoser

See Also

`kola.background`

Examples

```
### coming soon...
```

prepare *Function for tranformation and standardisation*

Description

This function can used for transformation and standardisation of the data.

Usage

```
prepare(x, scaling = "classical", transformation = "logarithm", powers = "none")
```

Arguments

x	data frame or matrix
scaling	Scaling of the data. Possible values are: "classical", "robust", "none"
transformation	Transformation of the data. Possible values are: "logarithm", "boxcox", "bcOpt", "logratio","logcentered","iso","none"
powers	Powers for Box-Cox transformation for each variable (if "boxcox" is chosen)

Details

Transformation:

"logarithm" replaces the values of x with the natural logarithm by using function 'log'.

"boxcox" apply a Box-Cox transformation on each variable. Powers must be specified.

"bcOpt" apply a Box-Cox transformation on each variable. Powers are calculated with function 'box.cox.powers'.

"none" is also possible.

Transformation before clustering: Cluster analysis in general does not need normally distributed data. However, it is advisable that heavily skewed data are first transformed to a more symmetric distribution. If a good cluster structure exists for a variable we can expect a distribution which has two or more modes. A transformation to more symmetry will preserve the modes but remove large skewness.

Standardisation:

"classical" apply a z-Transformation on each variable by using function 'scale'.

"robust" apply a robustified z-Transformation by using median and MAD.

"none" is also possible.

Standardisation before clustering: Standardisation is needed if the variables show a striking difference in the amount of variablity.

Value

Transformed and standardised data.

Author(s)

Matthias Templ

See Also

scale, box.cox.powers

Examples

```
require(mvoutlier)
data(humus)
x <- humus[,4:40]
xNew <- prepare(x, scaling="classical", transformation="logarithm")
```

---

print.clust *Print method for object clust*

Description

Print method for object 'clust'.

Usage

```
## S3 method for class 'clust'
print(x, ...)
```

Arguments

x	Object from function 'clust'
...	Additional arguments passed through

Author(s)

Matthias Templ

See Also

clust

Examples

```
GUIspatClust()
```

Index