Package 'tclust'

May 24, 2018

Version 1.4-1

Date 2018-05-24

Title Robust Trimmed Clustering

Author Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

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

Description Provides functions for robust trimmed clustering. The methods are described in Garcia-Escudero (2008) <doi:10.1214/07-AOS515>, Fritz et al. (2012) <doi:10.18637/jss.v047.i12> and others.

Depends R (>= 2.12.0)

Suggests mclust, cluster, mvtnorm, sn

License GPL-3

Repository CRAN

NeedsCompilation yes

Date/Publication 2018-05-24 21:50:22 UTC

R topics documented:

tclust-package *General Trimmed Cluster Analysis*

Description

A package implementing different (robust) clustering algorithms (`tclust`) based on trimming and including some graphical diagnostic tools (`ctlcurves` and `DiscrFact`)

Details

Package:	tclust
Type:	Package
Version:	1.0
Date:	2009-05-13
License:	GPL-3

Author(s)

Agustin Mayo Iscar, Heinrich Fritz, Maintainer: Luis Angel Garcia Escudero <lagarcia@eio.uva.es>

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2008), "A General Trimming Approach to Robust Cluster Analysis." Annals of Statistics, Vol.36, 1324-1345.
García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2010), "A Review of Robust Clustering Methods." Advances in Data Analysis and Classification, Vol.4, 89-109.
Fritz, H.; Garcia-Escudero, L.A.; Mayo-Iscar, A. (2012), "tclust: An R Package for a Trimming Approach to Cluster Analysis". Journal of Statistical Software, 47(12), 1-26. URL http://www.jstatsoft.org/v47/i12/

ctlcurves *Classification Trimmed Likelihood Curves*

Description

The function applies `tclust` several times on a given dataset while parameters `alpha` and `k` are altered. The resulting object gives an idea of the optimal trimming level and number of clusters considering a particular dataset.

Usage

```
ctlcurves (x, k = 1:4, alpha = seq (0, 0.2, len = 6),
           restr.fact = 50, trace = 1, ...)
```

Arguments

x	A matrix or data frame of dimension n x p, containing the observations (row-wise).
k	A vector of cluster numbers to be checked. By default cluster numbers from 1 to 5 are examined.
alpha	A vector containing the alpha levels to be checked. By default alpha levels from 0 to 0.2 (continuously increased by 0.01), are checked.
restr.fact	The restriction factor passed to tclust.
...	Further arguments (as e.g. restr), passed to tclust.
trace	Defines the tracing level, which is set to 1 by default. Tracing level 2 gives additional information on the current iteration.

Details

These curves show the values of the trimmed classification (log-)likelihoods when altering the trimming proportion alpha and the number of clusters k. The careful examination of these curves provides valuable information for choosing these parameters in a clustering problem. For instance, an appropriate k to be chosen is one that we do not observe a clear increase in the trimmed classification likelihood curve for k with respect to the k+1 curve for almost all the range of alpha values. Moreover, an appropriate choice of parameter alpha may be derived by determining where an initial fast increase of the trimmed classification likelihood curve stops for the final chosen k. A more detailed explanation can be found in García-Escudero et al. (2010).

Value

The function returns an S3 object of type ctlcurves with components:

par	A list containing all the parameters passed to this function.
obj	An array containing the objective functions values of each computed cluster-solution.
min.weights	An array containing the minimum cluster weight of each computed cluster-solution.

So far there is no output available for print.ctlcurves. Use plot on an ctlcurves object for a graphical interpretation of it.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2010), "Exploring the number of groups in robust model-based clustering." Statistics and Computing, (Forthcoming).
Preprint available at www.eio.uva.es/infor/personas/langel.html.

See Also

plot.ctlcurves

Examples

```
## Not run:
#--- EXAMPLE 1 ---------------------------------------

sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(108, cen * 0,   sig),
       mvtnorm::rmvnorm(162, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(30, cen * 2.5, sig * 50)
)

ctl <- ctlcurves (x, k = 1:4)

  ##  ctl-curves
plot (ctl)  ##  --> selecting k = 2, alpha = 0.08

  ##  the selected model
plot (tclust (x, k = 2, alpha = 0.08, restr.fact = 7))

#--- EXAMPLE 2 ---------------------------------------

data (geyser2)
ctl <- ctlcurves (geyser2, k = 1:5)

  ##  ctl-curves
plot (ctl)  ##  --> selecting k = 3, alpha = 0.08

  ##  the selected model
plot (tclust (geyser2, k = 3, alpha = 0.08, restr.fact = 5))


#--- EXAMPLE 3 ---------------------------------------

data (swissbank)
ctl <- ctlcurves (swissbank, k = 1:5, alpha = seq (0, 0.3, by = 0.025))

  ##  ctl-curves
plot (ctl)  ##  --> selecting k = 2, alpha = 0.1

  ##  the selected model
plot (tclust (swissbank, k = 2, alpha = 0.1, restr.fact = 50))

## End(Not run)
```

DiscrFact *Discriminant Factor Analysis for tclust Objects*

Description

Analyzes a `tclust`-object by calculating discriminant factors and comparing the quality of the actual cluster assignments and the second best possible assignment for each observation. Discriminant factors, measuring the strength of the "trimming" decision may also be defined. Cluster assignments of observations with large discriminant factors are considered as "doubtful" decisions. Silhouette plots give a graphical overview of the discriminant factors distribution (see `plot.DiscrFact`). More details can be found in García-Escudero et al. (2010).

Usage

```
DiscrFact(x, threshold = 1/10)
```

Arguments

`x`	A `tclust` object.
`threshold`	A cluster assignment or a trimming decision for an observation with a discriminant factor larger than `log(threshold)` is considered as a "doubtful" decision.

Details

This function compares the actual (best) assignment of each observation to its second best possible assignment. This comparison is based on the discriminant factors of each observation, which are calculated here. If the discriminant factor of an observation is larger than a given level (`log (threshold)`), the observation is considered as "doubtfully" assigned to a cluster. More information is shown when plotting the returned `DiscrFact` object.

Value

The function returns an S3 object of type `DiscrFact` containing the following components:

`x`	A `tclust` object.
`ylimmin`	A minimum y-limit calculated for plotting purposes.
`ind`	The actual cluster assignment.
`ind2`	The second most likely cluster assignment for each observation.
`disc`	The (weighted) likelihood of the actual cluster assignment of each observation.
`disc2`	The (weighted) likelihood of the second best cluster assignment of each observation.
`assignfact`	The factor `log (disc/disc2)`.
`threshold`	The threshold used for deciding whether `assignfact` indicates a "doubtful" assignment.
`mean.DiscrFact`	A vector of length `k + 1` containing the mean discriminant factors for each cluster (including the outliers).

Author(s)

Agustin Mayo-Iscar, Luis Angel García-Escudero, Heinrich Fritz

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2010), "Exploring the number of groups in robust model-based clustering." Statistics and Computing, (Forthcoming). Preprint available at www.eio.uva.es/infor/personas/langel.html.

See Also

plot.DiscrFact

Examples

```
sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
       mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
)
clus.1 <- tclust (x, k = 2, alpha = 0.1, restr.fact = 12)

clus.2 <- tclust (x, k = 3, alpha = 0.1, restr.fact = 1)
  ##  restr.fact and k are chosen improperly for pointing out the
  ##    difference in the plot of DiscrFact

dsc.1 <- DiscrFact (clus.1)
plot(dsc.1)

dsc.2 <- DiscrFact (clus.2)
plot (dsc.2)
```

---

discr_coords *Discriminant coordinates/canonical variates of tclust objects*

Description

Computes the two first discriminant coordinates (canonical coordinates) directly from a `tclust` object to obtain a graphical representations of cluster solutions in higher dimensional ($p > 2$) cases.

Usage

```
discr_coords(x, equal.weights)
```

Arguments

x	A tclust object.
equal.weights	A logical value, controlling whether the clusters should be considered as equal-sized (TRUE) when combining their covariance structures, or if their actual size shall be considered (FALSE). By default value xparequal.weights is assumed.

Details

The functionality of discr_coords is directly derived from discrcoord as implemented in the package "fpc" by Christian Hennig. It has been adopted in order to directly use the covariance information contained in the tclust-object. The function fails, if "store.x = FALSE" is specified in tclust, because the original data matrix is required here.

Value

A two-dimensional matrix, containing the canonical coordinates of all observations given by the tclust-object.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

Hennig, C. and Christlieb, N. (2002), "Validating visual clusters in large datasets: fixed point clusters of spectral features.", Computational Statistics and Data Analysis Vol.40, 723-739.

---

geyser2 *Old Faithful Geyser Data*

Description

A bivariate data set obtained from the Old Faithful Geyser, containing the eruption length and the length of the previous eruption for 271 eruptions of this geyser in minutes.

Usage

```
geyser2
```

Format

Eruption length The eruption length in minutes.

Previous eruption length The length of the previous eruption in minutes.

Source

This particular data structure can be obtained by applying the following code to the "Old Faithful Geyser" (`faithful` data set (Härdle 1991) in the package `datasets`):

```
f1 <- faithful[,1]
geyser2 <- cbind (f1[-length(f1)], f1[-1])
colnames (geyser2) <- c("Eruption length",
"Previous eruption length")
```

References

García-Escudero, L.A.; Gordaliza, A. (1999). "Robustness properties of k-means and trimmed k-means". Journal of the American Statistical Assoc., Vol.94, No.447, 956-969.
Härdle, W. (1991). "Smoothing Techniques with Implementation in S.", New York: Springer.

`M5data` *Mixture M5 Data*

Description

A bivariate data set obtained from three normal bivariate distributions with different scales and proportions 1:2:2. One of the components is very overlapped with another one. A 10% background noise is added uniformly distributed in a rectangle containing the three normal components and not very overlapped with the three mixture components. A precise description of the M5 data set can be found in García-Escudero et al. (2008).

Usage

```
M5data
```

Format

The first two columns are the two variables. The last column is the true classification vector where symbol "0" stands for the contaminating data points.

Source

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2008), "A General Trimming Approach to Robust Cluster Analysis". Annals of Statistics, Vol.36, pp. 1324-1345. Technical report available at http://www.eio.uva.es/inves/grupos/representaciones/trTCLUST.pdf

plot.ctlcurves *plot Method for ctlcurves Objects*

Description

The plot method for class ctlcurves: This function plots a ctlcurves object, comparing the target functions values with different values of parameter restr.fact.

Usage

```
## S3 method for class 'ctlcurves'
 plot(x, what = c ("obj", "min.weights", "doubtful"),
         main, xlab, ylab, xlim, ylim, col, lty = 1, ...)
```

Arguments

x	The ctlcurves object to be printed.
what	A string indicating which type of plot shall be drawn. See the details section for more information.
main	A character-string containing the title of the plot.
xlab, ylab, xlim, ylim	Arguments passed to plot.
col	A single value or vector of line colors passed to lines.
lty	A single value or vector of line types passed to lines.
...	Arguments to be passed to or from other methods.

Details

These curves show the values of the trimmed classification (log-)likelihoods when altering the trimming proportion alpha and the number of clusters k. The careful examination of these curves provides valuable information for choosing these parameters in a clustering problem. For instance, an appropriate k to be chosen is one that we do not observe a clear increase in the trimmed classification likelihood curve for k with respect to the k+1 curve for almost all the range of alpha values. Moreover, an appropriate choice of parameter alpha may be derived by determining where an initial fast increase of the trimmed classification likelihood curve stops for the final chosen k. A more detailed explanation can be found in García-Escudero et al. (2010).

This function implements a series of plots, which display characteristic values of the each model, computed with different values for k and alpha. The plot type is selected by setting argument what to one of the following values:

"obj" Objective function values.

"min.weights" The minimum cluster weight found for each computed model. This plot is intended to spot spurious clusters, which in general yield quite small weights.

"doubtful" The number of "doubtful" decisions identified by DiscrFact.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2010), "Exploring the number of groups in robust model-based clustering." Statistics and Computing, (Forthcoming). Preprint available at www.eio.uva.es/infor/personas/langel.html.

Examples

```
sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(108, cen * 0,   sig),
       mvtnorm::rmvnorm(162, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(30, cen * 2.5, sig * 50)
)

ctl <- ctlcurves(x, k = 1:4)
plot(ctl)
```

---

plot.DiscrFact *plot Method for DiscrFact Objects*

Description

The plot method for class DiscrFact: Next to a plot of the tclust object which has been used for creating the DiscrFact object, a silhouette plot indicates the presence of groups with a large amount of doubtfully assigned observations. A third plot similar to the standard tclust plot serves to highlight the identified doubtful observations.

Usage

```
## S3 method for class 'DiscrFact'
 plot(x, enum.plots = FALSE, ...)
plot_DiscrFact_p2 (x, xlab = "Discriminant Factor",
                   ylab = "Clusters", main, xlim,
                   print.Discr = TRUE, main.pre, ...)


plot_DiscrFact_p3 (x, main = "Doubtful Assignments", col, pch,
                   col.nodoubt = grey (0.8), by.cluster = FALSE,
                   ...)
```

Arguments

x	An object of class "DiscrFact" as from DiscrFact ().
enum.plots	A logical value indicating whether the plots shall be enumerated in their title ("(a)", "(b)", "(c)").
xlab, ylab, xlim	Arguments passed to funcion plot.tclust.
main	Argument passed to funcion plot.
print.Discr	A logical value indicating whether each clusters mean discriminant factor shall be plotted
main.pre	An optional string which is appended to the plot's caption.
pch, col	Arguments passed to function plot.
col.nodoubt	Color of all observations not considered as to be assigned doubtfully.
by.cluster	Logical value indicating whether parameters pch and col refer to observations (FALSE) or clusters (TRUE).
...	Arguments to be passed to or from other methods.

Details

plot.DiscrFact.p2 displays a silhouette plot based on the discriminant factors of the observations. A solution with many large discriminant factors is not reliable. Such clusters can be identified with this silhouette plot. Thus plot.DiscrFact.p3 displays the dataset, highlighting observations with discriminant factors greater than the given threshold. Function plot.DiscrFact combines the standard plot of a tclust object, and the two plots introduced here.

Value

No return value is provided.

Author(s)

Agustin Mayo Iscar, Luis Angel García Escudero, Heinrich Fritz

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2010), "Exploring the number of groups in robust model-based clustering." Statistics and Computing, (Forthcoming).
Preprint available at www.eio.uva.es/infor/personas/langel.html.

Examples

```
sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
       mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
)
```

```
clus.1 <- tclust (x, k = 2, alpha=0.1, restr.fact=12)
clus.2 <- tclust (x, k = 3, alpha=0.1, restr.fact=1)

dsc.1 <- DiscrFact (clus.1)
plot(dsc.1)

dsc.2 <- DiscrFact (clus.2)
plot (dsc.2)

dev.off ()
plot_DiscrFact_p2 (dsc.1)
plot_DiscrFact_p3 (dsc.2)
```

`plot.tclust` *plot Method for tclust Objects*

Description

The `plot` method for classes `tclust` and `tkmeans`.

Usage

```
## S3 method for class 'tclust'
 plot(x, ...)
## S3 method for class 'tkmeans'
 plot(x, ...)
```

Arguments

`x`	The `tclust` or `tkmeans` object to be displayed.
`...`	Further (optional) arguments which specify the details of the resulting plot (see section "Further Arguments").

Details

One and two dimensional structures are treated separately (e.g. tolerance intervals/ellipses are displayed). Higher dimensional structures are displayed by plotting the two first Fisher's canonical coordinates (evaluated by `discr_coords`) and derived from the final cluster assignments (trimmed observations are not taken into account). `plot.tclust.Nd` can be called with one or two-dimensional `tclust`-objects too. The function fails, if "`store.x = FALSE`" is specified in `tclust`, because the original data matrix is required here.

Further Arguments

`xlab, ylab, xlim, ylim, pch, col` Arguments passed to `plot`.

`main` The title of the plot. Use "/p" for displaying the chosen parameters `alpha` and `k` or "/r" for plotting the chosen restriction (`tclust` only).

`main.pre` An optional string which is added to the plot's caption.

`sub` A string specifying the subtitle of the plot. Use "/p" (default) for displaying the chosen parameters `alpha` and `k`, "/r" for plotting the chosen restriction (`tclust` only) and "/pr" for both.

`sub1` A secondary (optional) subtitle.

`labels` A string specifying the type of labels to be drawn. Either `"none"` (default), `"cluster"` or `"observation"` can be specified. If specified, parameter `pch` is ignored.

`text` A vector of length `n` (the number of observations) containing strings which are used as labels for each observation. If specified, the parameters `labels` and `pch` are ignored.

`by.cluster` Logical value indicating whether parameters `pch` and `col` refer to observations (`FALSE`) or clusters (`TRUE`).

`jitter.y` Logical value, specifying whether the drawn values shall be jittered in y-direction for better visibility of structures in 1 dimensional data.

`tol` The tolerance interval. 95% tolerance ellipsoids (assuming normality) are plotted by default (`tclust` only).

`tol.col, tol.lty, tol.lwd` Vectors of length `k` or `1` containing the `col`, `lty` and `lwd` arguments for the tolerance ellipses/lines (`tclust` only).

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

Examples

```
#--- EXAMPLE 1------------------------------
sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
       mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
)
# Two groups and 10% trimming level
a <- tclust (x, k = 2, alpha = 0.1, restr.fact = 12)

plot (a)
plot (a, labels = "observation")
plot (a, labels = "cluster")
plot (a, by.cluster = TRUE)

#--- EXAMPLE 2------------------------------
x <- c( rnorm(72,0,   1),
rnorm(108, 10,   1),
rnorm(20,  2.5, 10))

a <- tkmeans (x, k = 2, alpha = 0.1)
plot (a, jitter.y = TRUE)
```

summary.DiscrFact *summary Method for DiscrFact Objects*

Description

The summary method for class DiscrFact.

Usage

```
## S3 method for class 'DiscrFact'
 summary(object, hide.emtpy = TRUE, show.clust, show.alt, ...)
```

Arguments

object	An object of class "DiscrFact" as from DiscrFact ().
hide.emtpy	A logical value specifying whether clusters without doubtful assignment shall be hidden.
show.clust	A logical value specifying whether the number of doubtful assignments per cluster shall be displayed.
show.alt	A logical value specifying whether the alternative cluster assignment shall be displayed.
...	Arguments passed to or from other methods.

Value

No return value is provided.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

García-Escudero, L.A.; Gordaliza, A.; Matrán, C. and Mayo-Iscar, A. (2009), "Exploring the number of groups in robust model-based clustering".
Preprint available at www.eio.uva.es/infor/personas/langel.html.

See Also

plot.DiscrFact

Examples

```
sig <- diag (2)
cen <- rep (1, 2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
       mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
       mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
)
clus.1 <- tclust (x, k = 2, alpha = 0.1, restr.fact = 12)

clus.2 <- tclust (x, k = 3, alpha = 0.1, restr.fact = 1)
  ##  restr.fact and k are chosen improperly for pointing out the
  ##    difference in the plot of DiscrFact

dsc.1 <- DiscrFact (clus.1)
summary(dsc.1)

dsc.2 <- DiscrFact (clus.2)
summary (dsc.2)
```

---

swissbank *SwissBankNotes Data*

Description

Six variables measured on 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes (Flury and Riedwyl, 1988).

Usage

```
swissbank
```

Format

`Length` Length of the bank note

`Ht_Left` Height of the bank note, measured on the left

`Ht_Right` Height of the bank note, measured on the right

`IF_Lower` Distance of inner frame to the lower border

`IF_Upper` Distance of inner frame to the upper border

`Diagonal` Length of the diagonal

Details

Observations 1-100 are the genuine bank notes and the other 100 observations are the counterfeit bank notes.

Source

Flury, B. and Riedwyl, H. (1988). Multivariate Statistics, A Practical Approach, Cambridge University Press.

`tclust` *General Trimming Approach to Robust Cluster Analysis*

Description

`tclust` searches for `k` (or less) clusters with different covariance structures in a data matrix `x`. Relative cluster scatter can be restricted by a constant value `restr.fact`. For robustifying the estimation, a proportion `alpha` of observations may be trimmed.

In particular, the trimmed k-means method (`tkmeans`)is represented by the `tclust` method, setting parameters `restr = "eigen"`, `restr.fact = 1` and `equal.weights = TRUE`.

Usage

```
tclust (x, k = 3, alpha = 0.05, nstart = 50, iter.max = 20,
        restr = c ("eigen", "deter", "sigma"), restr.fact = 12,
        equal.weights = FALSE, center, scale, store.x = TRUE,
        drop.empty.clust = TRUE, trace = 0, warnings = 3,
        zero.tol = 1e-16)
```

Arguments

`x`	A matrix or data.frame of dimension `n` x `p`, containing the observations (row-wise).
`k`	The number of clusters initially searched for.
`alpha`	The proportion of observations to be trimmed.
`nstart`	The number of random initializations to be performed.
`iter.max`	The maximum number of concentration steps to be performed. The concentration steps are stopped, whenever two consecutive steps lead to the same data partition.
`restr`	The type of restriction to be applied on the cluster scatter matrices. Valid values are `"eigen"` (default), `"deter"` and `"sigma"`. See the detail section for further explanation.
`restr.fact`	The constant `restr.fact >= 1` constrains the allowed differences among group scatters. Larger values imply larger differences of group scatters, a value of 1 specifies the strongest restriction. When using `restr = "sigma"` this parameter is not considered, as all cluster variances are averaged, always implying `restr.fact = 1`.

`equal.weights` A logical value, specifying whether equal cluster weights (`TRUE`) or not (`FALSE`) shall be considered in the concentration and assignment steps.

`center, scale` A center and scale vector, each of length `p` which can optionally be specified for centering and scaling `x` before calculation

`store.x` A logical value, specifying whether the data matrix `x` shall be included in the result structure. By default this value is set to `TRUE`, because functions `plot.tclust` and `DiscrFact` depend on this information. However, when big data matrices are handled, the result structure's size can be decreased noticeably when setting this parameter to `FALSE`.

`drop.empty.clust` Logical value specifying, whether empty clusters shall be omitted in the resulting object. (The result structure does not contain center and covariance estimates of empty clusters anymore. Cluster names are reassigned such that the first `l` clusters (`l <= k`) always have at least one observation.

`trace` Defines the tracing level, which is set to `0` by default. Tracing level `2` gives additional information on the iteratively decreasing objective function's value.

`warnings` The warning level (0: no warnings; 1: warnings on unexpected behavior; 2: warnings if `restr.fact` causes artificially restricted results).

`zero.tol` The zero tolerance used. By default set to 1e-16.

Details

This iterative algorithm initializes `k` clusters randomly and performs "concentration steps" in order to improve the current cluster assignment. The number of maximum concentration steps to be performed is given by `iter.max`. For approximately obtaining the global optimum, the system is initialized `nstart` times and concentration steps are performed until convergence or `iter.max` is reached. When processing more complex data sets higher values of `nstart` and `iter.max` have to be specified (obviously implying extra computation time). However, if more then half of the iterations would not converge, a warning message is issued, indicating that `nstart` has to be increased.

The parameter `restr` defines the cluster's shape restrictions, which are applied on all clusters during each iteration. Options `"eigen"`/`"deter"` restrict the ratio between the maximum and minimum eigenvalue/determinant of all cluster's covariance structures to parameter `restr.fact`. Setting `restr.fact` to `1`, yields the strongest restriction, forcing all eigenvalues/determinants to be equal and so the method looks for similarly scattered (respectively spherical) clusters. Option `"sigma"` is a simpler restriction, which averages the covariance structures during each iteration (weighted by cluster sizes) in order to get similar (equal) cluster scatters.

Value

The function returns an S3 object of type `tclust`, containing the following values:

`centers` A matrix of size `p` x `k` containing the centers (column-wise) of each cluster.

`cov` An array of size `p` x `p` x `k` containing the covariance matrices of each cluster.

`cluster` A numerical vector of size `n` containing the cluster assignment for each observation. Cluster names are integer numbers from `1` to `k`, `0` indicates trimmed observations.

par	A list, containing the parameters the algorithm has been called with (x, if not suppressed by store.x = FALSE, k, alpha, restr.fact, nstart, KStep, and equal.weights).
k	The (final) resulting number of clusters. Some solutions with a smaller number of clusters might be found when using the option equal.weights = FALSE.
obj	The value of the objective function of the best (returned) solution.
size	An integer vector of size k, returning the number of observations contained by each cluster.
weights	A numerical vector of length k, containing the weights of each cluster.
int	A list of values internally used by function related to tclust objects.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

Garcia-Escudero, L.A.; Gordaliza, A.; Matran, C. and Mayo-Iscar, A. (2008), "A General Trimming Approach to Robust Cluster Analysis". Annals of Statistics, Vol.36, 1324-1345. Technical Report available at www.eio.uva.es/inves/grupos/representaciones/trTCLUST.pdf

Fritz, H.; Garcia-Escudero, L.A.; Mayo-Iscar, A. (2012), "tclust: An R Package for a Trimming Approach to Cluster Analysis". Journal of Statistical Software, 47(12), 1-26. URL http://www.jstatsoft.org/v47/i12/

Examples

```
#--- EXAMPLE 1 ---------------------------------------
sig <- diag (2)
cen <- rep (1,2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
           mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
           mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
           )

# Two groups and 10% trimming level
clus <- tclust (x, k = 2, alpha = 0.1, restr.fact = 8)

plot (clus)
plot (clus, labels = "observation")
plot (clus, labels = "cluster")

# Three groups (one of them very scattered) and 0% trimming level
clus <- tclust (x, k = 3, alpha=0.0, restr.fact = 100)

plot (clus)
```

```
#--- EXAMPLE 3 ----------------------------------------
data (M5data)
x <- M5data[, 1:2]

clus.a <- tclust (x, k = 3, alpha = 0.1, restr.fact =  1,
                  restr = "eigen", equal.weights = TRUE, warnings = 1)
clus.b <- tclust (x, k = 3, alpha = 0.1, restr.fact =  1,
                   equal.weights = TRUE, warnings = 1)
clus.c <- tclust (x, k = 3, alpha = 0.1, restr.fact =  1,
                  restr = "deter", equal.weights = TRUE, iter.max = 100,
  warnings = 1)
clus.d <- tclust (x, k = 3, alpha = 0.1, restr.fact = 50,
                  restr = "eigen", equal.weights = FALSE)

pa <- par (mfrow = c (2, 2))
plot (clus.a, main = "(a) tkmeans")
plot (clus.b, main = "(b) Gallegos and Ritter")
plot (clus.c, main = "(c) Gallegos")
plot (clus.d, main = "(d) tclust")
par (pa)

#--- EXAMPLE 4 ----------------------------------------
data (swissbank)
# Two clusters and 8% trimming level
clus <- tclust (swissbank, k = 2, alpha = 0.08, restr.fact = 50)

                            # Pairs plot of the clustering solution
pairs (swissbank, col = clus$cluster + 1)
                                  # Two coordinates
plot (swissbank[, 4], swissbank[, 6], col = clus$cluster + 1,
     xlab = "Distance of the inner frame to lower border",
     ylab = "Length of the diagonal")
plot (clus)

# Three clusters and 0% trimming level
clus <- tclust (swissbank, k = 3, alpha = 0.0, restr.fact = 110)

                            # Pairs plot of the clustering solution
pairs (swissbank, col = clus$cluster + 1)

                                   # Two coordinates
plot (swissbank[, 4], swissbank[, 6], col = clus$cluster + 1,
      xlab = "Distance of the inner frame to lower border",
      ylab = "Length of the diagonal")

plot (clus)
```

---

tkmeans *Trimmed k-means Cluster Analysis*

Description

tkmeans searches for k (or less) spherical clusters in a data matrix x, whereas the ceiling (alpha n) most outlying observations are trimmed.

Usage

```
tkmeans (x, k = 3, alpha = 0.05, nstart = 50, iter.max = 20,
         equal.weights = FALSE, center = 0, scale = 1, store.x = TRUE,
         drop.empty.clust = TRUE, trace = 0, warnings = 2, zero.tol = 1e-16)
```

Arguments

x	A matrix or data.frame of dimension n x p, containing the observations (row-wise).
k	The number of clusters initially searched for.
alpha	The proportion of observations to be trimmed.
nstart	The number of random initializations to be performed.
iter.max	The maximum number of concentration steps to be performed. The concentration steps are stopped, whenever two consecutive steps lead to the same data partition.
equal.weights	A logical value, specifying whether equal cluster weights (TRUE) or not (FALSE) shall be considered in the concentration and assignment steps.
center, scale	A center and scale vector, each of length p which can optionally be specified for centering and scaling x before calculation
store.x	A logical value, specifying whether the data matrix x shall be included in the result structure. By default this value is set to TRUE, because functions plot.tkmeans depends on this information. However, when big data matrices are handled, the result structure's size can be decreased noticeably when setting this parameter to FALSE.
drop.empty.clust	Logical value specifying, whether empty clusters shall be omitted in the resulting object. (The result structure does not contain center and covariance estimates of empty clusters anymore. Cluster names are reassigned such that the first l clusters (l <= k) always have at least one observation.
trace	Defines the tracing level, which is set to 0 by default. Tracing level 2 gives additional information on the iteratively decreasing objective function's value.
warnings	The warning level (0: no warnings; 1: warnings on unexpected behavior.
zero.tol	The zero tolerance used. By default set to 1e-16.

Value

The function returns an S3 object of type tkmeans, containing the following values:

centers	A matrix of size p x k containing the centers (column-wise) of each cluster.

cluster	A numerical vector of size n containing the cluster assignment for each observation. Cluster names are integer numbers from 1 to k, 0 indicates trimmed observations.
par	A list, containing the parameters the algorithm has been called with (x, if not suppressed by store.x = FALSE, k, alpha, restr.fact, nstart, KStep, and equal.weights).
k	The (final) resulting number of clusters. Some solutions with a smaller number of clusters might be found when using the option equal.weights = FALSE.
obj	The value of the objective function of the best (returned) solution.
size	An integer vector of size k, returning the number of observations contained by each cluster.
weights	A numerical vector of length k, containing the weights of each cluster.
int	A list of values internally used by function related to tkmeans objects.

Author(s)

Agustin Mayo Iscar, Luis Angel Garcia Escudero, Heinrich Fritz

References

Cuesta-Albertos, J. A.; Gordaliza, A. and Matrán, C. (1997), "Trimmed k-means: an attempt to robustify quantizers". Annals of Statistics, Vol. 25 (2), 553-576.

Examples

```
#--- EXAMPLE 1 ---------------------------------------
sig <- diag (2)
cen <- rep (1,2)
x <- rbind(mvtnorm::rmvnorm(360, cen * 0,   sig),
            mvtnorm::rmvnorm(540, cen * 5,   sig * 6 - 2),
            mvtnorm::rmvnorm(100, cen * 2.5, sig * 50)
            )

# Two groups and 10% trimming level
clus <- tkmeans (x, k = 2, alpha = 0.1)

plot (clus)
plot (clus, labels = "observation")
plot (clus, labels = "cluster")

#--- EXAMPLE 2 ---------------------------------------
data (geyser2)
clus <- tkmeans (geyser2, k = 3, alpha = 0.03)
plot (clus)

#--- EXAMPLE 3 ---------------------------------------
data (swissbank)
# Two clusters and 8% trimming level
```

```
clus <- tkmeans (swissbank, k = 2, alpha = 0.08)

                            # Pairs plot of the clustering solution
pairs (swissbank, col = clus$cluster + 1)
                                  # Two coordinates
plot (swissbank[, 4], swissbank[, 6], col = clus$cluster + 1,
     xlab = "Distance of the inner frame to lower border",
     ylab = "Length of the diagonal")
plot (clus)

# Three clusters and 0% trimming level
clus <- tkmeans (swissbank, k = 3, alpha = 0.0)

                            # Pairs plot of the clustering solution
pairs (swissbank, col = clus$cluster + 1)

                                   # Two coordinates
plot (swissbank[, 4], swissbank[, 6], col = clus$cluster + 1,
      xlab = "Distance of the inner frame to lower border",
      ylab = "Length of the diagonal")

plot (clus)
```

Index