

Package ‘lga’

April 17, 2009

Version 1.1-1

Date 2008-06-15

Title Tools for linear grouping analysis (LGA)

Author Justin Harrington

Maintainer Justin Harrington <harringt@stat.ubc.ca>

Depends R (>= 2.2.1)

Imports boot, lattice

Suggests snow

Description Tools for linear grouping analysis. Three user-level functions: gap, rlga and lga.

License GPL

Repository CRAN

Date/Publication 2008-07-20 09:09:11

R topics documented:

lga-package	2
brain	2
corridorWalls	3
gap	3
lga	6
nh194	9
ob	9

Index	11
--------------	-----------

lga-package

The lga Package

Description

The lga package is an implementation of the algorithms described in *Van Aelst et al (2006)* and *Garcia-Escudero et al (2008)*. It has three main functions (with accompanying print and plot methods).

`lga` The core linear grouping analysis

`rlga` The robust version of linear grouping analysis

`gap` Performs a gap analysis to find the number of clusters

For more details refer to the relevant help files.

Author(s)

Justin Harrington (harringt@stat.ubc.ca)

References

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) ‘Linear Grouping Using Orthogonal Regression’, *Computational Statistics & Data Analysis* **50**, 1287–1312.

Garcia-Escudero, L.A., Gordaliza, A., San Martin, R., Van Aelst, S. and Zamar, R.H. (2008) ‘Robust linear clustering’. To appear in *Journal of the Royal Statistical Society, Series B* (accepted June, 2008).

brain

Allometry

Description

Allometry data from *Van Aelst et al (2006)*

Usage

```
data(brain)
```

Format

A matrix with 282 observations on the following 2 variables:

BrainWeight.g. a numeric vector

OlfactoryBulbs.ml. a numeric vector

References

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) 'Linear Grouping Using Orthogonal Regression', *Computational Statistics & Data Analysis* **50**, 1287–1312.

corridorWalls	<i>Corridor Walls data</i>
---------------	----------------------------

Description

Corridor Walls data from Garcia-Escudero et al. (2008). A laser device is introduced in a corridor of an office building and throws a laser ray which touches a point from the object found at the end of its trajectory. The device produces a three dimensional measurement of the placement of that point with respect to a fixed reference system.

Usage

```
data(corridorWalls)
```

Format

A matrix with 11710 observations on the following 3 variables (3-dimensional coordinates).

x a numeric vector

y a numeric vector

z a numeric vector

References

Garcia-Escudero, L.A., Gordaliza, A., San Martin, R., Van Aelst, S. and Zamar, R.H. (2008) 'Robust linear clustering'. To appear in *Journal of the Royal Statistical Society, Series B* (accepted June, 2008)

gap	<i>Perform gap analysis</i>
-----	-----------------------------

Description

Performs the gap analysis using lga to estimate the number of clusters.

Usage

```
## Default S3 method:  
gap(x, K, B, criteria=c("tibshirani", "DandF", "none"),  
     nnode=NULL, scale=TRUE, ...)
```

Arguments

<code>x</code>	a numeric matrix.
<code>K</code>	an integer giving the maximum number of clusters to consider.
<code>B</code>	an integer giving the number of bootstraps.
<code>criteria</code>	a character string indicating which criteria to evaluate the gap data. One of “tibshirani” (default), “DandF” or “none”. Can be abbreviated.
<code>nnode</code>	an integer of many CPUS to use for parallel processing. Defaults to NULL i.e. no parallel processing.
<code>scale</code>	logical. Should the data be scaled?
<code>...</code>	For any other arguments passed from the generic function.

Details

This code performs the gap analysis using lga. The gap statistic is defined as the difference between the log of the Residual Orthogonal Sum of Squared Distances (denoted $\log(W_k)$) and its expected value derived using bootstrapping under the null hypothesis that there is only one cluster. In this implementation, the reference distribution used for the bootstrapping is a random uniform hypercube, transformed by the principal components of the underlying data set. For further details see *Tibshirani et al (2001)*.

For different criteria, different rules apply. With “tibshirani” (*ibid*) we calculate the gap statistic for $k = 1, \dots, K$, stopping when

$$\text{gap}(k) \geq \text{gap}(k + 1) - s_{k+1}$$

where s_{k+1} is a function of standard deviation of the bootstrapped estimates.

With the “DandF” criteria from *Dudoit et al (2002)*, we calculate the gap statistic for all values of $k = 1, \dots, K$, selecting the number of clusters as

$$\hat{k} = \text{smallest } k \geq 1 \text{ such that } \text{gap}(k) \geq \text{gap}(k^*) - s_{k^*}$$

where $k^* = \arg \max_{k \geq 1} \text{gap}(k)$.

Finally, for the criteria “none”, no rules are applied, and just the gap data is returned.

As lga is ostensibly unsupervised in this case, the parameter niter is set to 20 to ensure convergence.

This function is parallel computing aware via the `nnode` argument, and works with the package `snow`. In order to use parallel computing, one of MPI (e.g. `lamboot`) or PVM is necessary. For further details, see the documentation for `snow`.

Value

An object of class “gap” with components

<code>finished</code>	a logical. For the “tibshirani”, was there a solution found?
<code>nclust</code>	a integer for the number of clusters estimated. Returns NA if nothing conclusive is found.
<code>data</code>	the original data set, scaled if specified in the arguments.
<code>criteria</code>	the criteria used.

Author(s)

Justin Harrington (harringt@stat.ubc.ca)

References

Tibshirani, R. and Walther, G. and Hastie, T. (2001) ‘Estimating the number of clusters in a data set via the gap statistic’, *J. R. Statist. Soc. B* **63**, 411–423.

Dudoit, S. and Fridlyand, J. (2002) ‘A prediction-based resampling method for estimating the number of clusters in a dataset’, *Genome Biology* **3**.

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) ‘Linear Grouping Using Orthogonal Regression’, *Computational Statistics & Data Analysis* **50**, 1287–1312.

See Also

[lga](#)

Examples

```
## Synthetic example
## Make a dataset with 2 clusters in 2 dimensions

library(MASS)
set.seed(1234)
X <- rbind(mvrnorm(n=100, mu=c(1, -2), Sigma=diag(0.1, 2) + 0.9),
           mvrnorm(n=100, mu=c(1, 1), Sigma=diag(0.1, 2) + 0.9))

gap(X, K=4, B=20)

## to run this using parallel processing with 4 nodes, the equivalent
## code would be

## Not run: gap(X, K=4, B=20, nnode=4)

## Quakes data (from package:datasets)
## Including the first two dimensions versus three dimensions
## yields different results

set.seed(1234)
## Not run:
gap(quakes[,1:2], K=4, B=20)
gap(quakes[,1:3], K=4, B=20)
## End(Not run)

library(maps)
lgaout1 <- lga(quakes[,1:2], k=3)
plot(lgaout1)

lgaout2 <- lga(quakes[,1:3], k=2)
plot(lgaout2)
```

```
## Let's put this in context
par(mfrow=c(1,2))
map("world", xlim=range(quakes[,2]), ylim=range(quakes[,1])); box()
points(quakes[,2], quakes[,1], pch=lgaout1$cluster, col=lgaout1$cluster)

map("world", xlim=range(quakes[,2]), ylim=range(quakes[,1])); box()
points(quakes[,2], quakes[,1], pch=lgaout2$cluster, col=lgaout2$cluster)
```

lga

Perform LGA/RLGA

Description

Linear Grouping Analysis

Usage

```
## Default S3 method:
lga(x, k, biter = NULL, niter = 10, showall = FALSE, scale = TRUE,
    nnode=NULL, silent=FALSE, ...)
## Default S3 method:
rlga(x, k, alpha=0.9, biter = NULL, niter = 10, showall = FALSE, scale = TRUE,
    nnode=NULL, silent=FALSE, ...)
```

Arguments

x	a numeric matrix.
k	an integer for the number of clusters.
alpha	a numeric value between 0.5 and 1. For the robust estimate of LGA, specifying the percentage of points in the best subset.
biter	an integer for the number of different starting hyperplanes to try.
niter	an integer for the number of iterations to attempt for convergence.
showall	logical. If TRUE then display all the outcomes, not just the best one.
scale	logical. Allows you to scale the data, dividing each column by its standard deviation, before fitting.
nnode	an integer of many CPUS to use for parallel processing. Defaults to NULL i.e. no parallel processing.
silent	logical. If TRUE, produces no text output during processing.
...	For any other arguments passed from the generic function.

Details

This code tries to find `k` clusters using the lga algorithm described in *Van Aelst et al (2006)*. For each attempt, it has up to `niter` steps to get to convergence, and it does this from `biter` different starting hyperplanes. It then selects the clustering with the smallest Residual Orthogonal Sum of Squareds.

If `biter` is left as `NULL`, then it is selected via the equation given in *Van Aelst et al (2006)*.

The function `rlga` is the robust equivalent to LGA, and is introduced in *Garcia-Escudero et al (2008)*.

Both functions are parallel computing aware via the `nnode` argument, and works with the package `snow`. In order to use parallel computing, one of MPI (e.g. `lamboot`) or PVM is necessary. For further details, see the documentation for `snow`.

Associated with the `lga` and `rlga` functions are a print method and a plot method (see the examples). In the plot method, the fitted hyperplanes are also shown as dashed-lines when there are only two dimensions.

Value

An object of class “lga”. The list contains

<code>cluster</code>	a vector containing the cluster memberships.
<code>ROSS</code>	the Residual Orthogonal Sum of Squares for the solution.
<code>converged</code>	a logical. True if at least one solution has converged.
<code>nconverg</code>	the number of converged solutions (out of <code>biter</code> starts).
<code>x</code>	the (scaled if selected) dataset.
<code>scaled</code>	logical. Is the data scaled?
<code>k</code>	the number of clusters to be found.
<code>biter</code>	the <code>biter</code> setting used.
<code>niter</code>	the <code>niter</code> setting used.

Author(s)

Justin Harrington (harringt@stat.ubc.ca)

References

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) ‘Linear Grouping Using Orthogonal Regression’, *Computational Statistics & Data Analysis* **50**, 1287–1312.

Garcia-Escudero, L.A., Gordaliza, A., San Martin, R., Van Aelst, S. and Zamar, R.H. (2008) ‘Robust linear clustering’. To appear in *Journal of the Royal Statistical Society, Series B* (accepted June, 2008).

See Also

[gap](#)

Examples

```

## Synthetic Data
## Make a dataset with 2 clusters in 2 dimensions

library(MASS)
set.seed(1234)
X <- rbind(mvrnorm(n=100, mu=c(1,-1), Sigma=diag(0.1,2)+0.9),
           mvrnorm(n=100, mu=c(1,1), Sigma=diag(0.1,2)+0.9))

lgaout <- lga(X,2)
plot(lgaout)
print(lgaout)

## Robust equivalent

rlgaout <- rlga(X,2, alpha=0.75)
plot(rlgaout)
print(rlgaout)

## nhl94 data set

data(nhl94)
plot(lga(nhl94, k=3, niter=30))

## Allometry data set
data(brain)
plot(lga(log(brain, base=10), k=3))

## Second Allometry data set
data(ob)
plot(lga(log(ob[,2:3]), k=3), pch=as.character(ob[,1]))

## Corridor Walls data set
## To obtain the results reported in Garcia-Escudero et al. (2008):
data(corridorWalls)
rlgaout <- rlga(corridorWalls, k=3, biter = 100, niter = 30, alpha=0.85)
pairs(corridorWalls, col=rlgaout$cluster+1)
plot(rlgaout)

## Parallel processing case
## In this example, running using 4 nodes.

## Not run:
set.seed(1234)
X <- rbind(mvrnorm(n=1e6, mu=c(1,-1), Sigma=diag(0.1,2)+0.9),
           mvrnorm(n=1e6, mu=c(1,1), Sigma=diag(0.1,2)+0.9))
abc <- lga(X, k=2, nnode=4)
## End(Not run)

```

`nhl94`*Player Performance in NHL for 1994-1995*

Description

This data set gives four variables for each of 871 players for the NHL 1994-1995 season. From *Van Aelst et al (2006)*

Usage

```
data(nhl94)
```

Format

A matrix with 871 observations on the following 3 variables:

PTS Points scored, a numeric vector

PM plus/minus average rating, a numeric vector

PIM Total penalty time in minutes, a numeric vector

PP Power play goals, a numeric vector

References

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) ‘Linear Grouping Using Orthogonal Regression’, *Computational Statistics & Data Analysis* **50**, 1287–1312.

`ob`*Allometry Data*

Description

Allometry data from *Van Aelst et al (2006)*

Usage

```
data(ob)
```

Format

A data frame with 83 observations on the following 3 variables.

Group a factor with levels `a c h H i m p t`

BrainWeight.g. a numeric vector

OlfactoryBulbs.ml. a numeric vector

References

Van Aelst, S. and Wang, X. and Zamar, R. and Zhu, R. (2006) 'Linear Grouping Using Orthogonal Regression', *Computational Statistics & Data Analysis* **50**, 1287–1312.

Index

*Topic **cluster**

gap, [3](#)

lga, [6](#)

lga-package, [1](#)

*Topic **datasets**

brain, [2](#)

corridorWalls, [3](#)

nhl94, [9](#)

ob, [9](#)

*Topic **multivariate**

gap, [3](#)

lga, [6](#)

lga-package, [1](#)

brain, [2](#)

corridorWalls, [3](#)

gap, [1](#), [3](#), [7](#)

lga, [1](#), [5](#), [6](#)

lga-package, [1](#)

nhl94, [9](#)

ob, [9](#)

rlga, [1](#)

rlga(*lga*), [6](#)