Package 'multichull'

October 13, 2022

Type Package

Title A Generic Convex-Hull-Based Model Selection Method

Version 1.0.0

Date 2017-11-16

Depends igraph, graphics, plotly, shiny, shinythemes, stats

Description Given a set of models for which a measure of model (mis)fit and model complexity is provided, CHull(), developed by Ceulemans and Kiers (2006) <doi:10.1348/000711005X64817>, determines the models that are located on the boundary of the convex hull and selects an optimal model by means of the scree test values.

License GPL (>= 2)

RoxygenNote 6.0.1

NeedsCompilation no

Author Marlies Vervloet [aut, cre, trl],
Tom Wilderjans [aut],
Jeffrey Durieux [aut],
Eva Ceulemans [aut]

Maintainer Marlies Vervloet <`marlies.vervloet@kuleuven.be`>

Repository CRAN

Date/Publication 2017-11-21 18:19:11 UTC

R topics documented:

`multichull-package` *A Generic Convex-Hull-Based Model Selection Method*

Description

Given a set of models for which a measure of model (mis)fit and model complexity is provided, CHull() determines the models that are located on the boundary of the convex hull and selects an optimal model by means of the scree test values.

Details

Package:	**multichull**
Type:	Package
Version:	1.0.0
Date:	2017-11-16
License:	GPL (>= 2)

Author(s)

Marlies Vervloet (<`marlies.vervloet@kuleuven.be`>)

References

Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013). CHull: A generic convex hull based model selection method. Behavior Research Methods, 45, 1-15.

Ceulemans, E., & Kiers, H. A. L. (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. British Journal of Mathematical & Statistical Psychology, 59, 133-150.

See Also

`CHull` `MultiCHull`

Examples

```
complexity.fit <- cbind(c(305,456,460,607,612,615,758,764,768,770,909,916,921,924),
c(152,89,79,71,57,57,64,49,47,47,60,41,39,39))

output <- CHull(complexity.fit)
plot(output)
print(output)
summary(output)
```

`CHull` *A Generic Convex-Hull-Based Model Selection Method*

Description

Given a set of models for which a measure of model (mis)fit and model complexity is provided, CHull determines the models that are located on the boundary of the convex hull and selects an optimal model by means of the scree test values.

Usage

```
CHull(data, bound = "lower", PercentageFit = 0.01)

## S3 method for class 'CHull'
plot(x, col = NULL, pch = NULL, ...)

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

## S3 method for class 'CHull'
summary(object, ...)
```

Arguments

`data`	Dataframe with complexity in 1st column and fit measures in 2nd column
`bound`	Boundary of convex hull to inspect: `upper` or `lower`
`PercentageFit`	Required proportion of increase in fit of a more complex model
`x`	An object of the type produced by `CHull`
`col`	Vector of `colors` used for plots
`pch`	Symbol used to indicate selected model(s)
`...`	Additional arguments
`object`	An object of the type produced by `CHull`

Value

`Solution`	Dataframe with selected models
`Hull`	Dataframe with all models on hull boundary and their *st* value
`Origdata`	Original dataframe
`Bound`	Boundary of convex hull that was requested
`PercentageFit`	Requested proportion of increase in fit of a more complex model

Details

The CHull method (Wilderjans, Ceulemans, & Meers, 2013) can be used for selecting a model by comparing model complexities (1st column of the input parameter `data`) and fit values (2nd column).

In a first step, only the best model (or one of the best, if some models have an equal fit) is retained per complexity. This should be a model with a high fit value if the fit measure indicates goodness-of-fit (`bound="upper"`) and a low fit value if it indicates badness-of-fit (`bound="upper"`). A warning will be generated if the sign of the correlation between complexity and fit is counterintuitive in this regard.

In a second step, the remaining models are ordered (increasingly) on the basis of their complexity value.

In Step 3, models are excluded that have a higher complexity but a worse (or equal) fit when compared to the other models. This procedure is repeated until a monotonical increase (`bound="upper"`) or decrease (`bound="lower"`) is reached. If less than 3 models remain, the method generates the error warning that not enough data points are available for computing the convex hull and the procedure stops.

In Step 4, it is determined which models lie on the `upper` or `lower` boundary of the convex hull. Models are discarded if the improvement in fit, compared to a less complex model, is less than `PercentageFit` (default: `PercentageFit=.01`). The remaining models are returned in `Hull`.

Step 5 consists of computing the scree test values (`st`) for each remaining model. This is, however, not possible for the most simple and most complex model, and those models will therefore never be selected as the optimal solution, except when these are the only models that remain after the previous step.

In Step 6, the model with highest scree test value is selected, and finally, in Step 7, also models are selected that were excluded in the first step, but that have the same complexity and fit value as the selected model. All selected models end up in `Solution`.

References

Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013). CHull: A generic convex hull based model selection method. Behavior Research Methods, 45, 1-15.

Ceulemans, E., & Kiers, H. A. L. (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. British Journal of Mathematical & Statistical Psychology, 59, 133-150.

Examples

```
complexity.fit <- cbind(c(305,456,460,607,612,615,758,764,768,770,909,916,921,924),
c(152,89,79,71,57,57,64,49,47,47,60,41,39,39))

output <- CHull(complexity.fit)
plot(output)
print(output)
summary(output)
```

`MultiCHull` *Convex-Hull-Based Model Selection for multiple Samples*

Description

Applying the `CHull` function on multiple samples of fit values at once, such as bootstrap samples.

Usage

```
MultiCHull(data, bound = "lower", PercentageFit = 0.01)

## S3 method for class 'MultiCHull'
plot(x, col = NULL, pch = NULL, whichticks = NULL,
  las = 2, ...)

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

## S3 method for class 'MultiCHull'
summary(object, ...)
```

Arguments

`data`	Dataframe with complexity in 1st column and fit measures in next columns
`bound`	Boundary of convex hull to inspect: `upper` or `lower`
`PercentageFit`	Required proportion of increase in fit of a more complex model
`x`	An object of the type produced by `MultiCHull`
`col`	Vector of `colors` used for plots
`pch`	Vector of `pch` symbols
`whichticks`	Model names of ticks that should be displayed
`las`	Orientation of tick mark labels
`...`	Additional arguments
`object`	An object of the type produced by `MultiCHull`

Value

`st`	Dataframe with scree test values
`tab`	Table which indicates the selected model in each sample
`frq`	Table which indicates how often each model is selected
`Origdata`	Original dataframe
`Bound`	Boundary of convex hull that was requested
`PercentageFit`	Requested proportion of increase in fit of a more complex model

Details

`MultiCHull` **function:** `MultiCHull` applies the `CHull` code on multiple samples of fit values. To this end, the input parameter `data` consists of a dataframe with complexity values in the first column and fit values in the next columns. The different samples can for example be bootstrap samples, or fit values obtained with different random starts, or from different fit measures, etc. It is possible that in some samples no optimal solution can be found. This will generate a warning, which will include the sample number.

Data frame `st` contains per sample the scree test values of the solutions that were found on the `upper` or `lower` bound of the hull (see also `CHull`). In each sample, the least and most complex model receive a 0 value. The other models have an NA value. `tab` is also a dataframe, which indicates per sample the top three of optimal models (indicated by a 1, 2 and 3). The other models have an NA value. Finally, in `frq` the frequencies are shown for each model of being selected as the optimal model.

Plot function: Applying the method `plot()` on output of `MultiCHull` yields a plot with the models on the x-axis, ordered by increasing complexity. By default, all model names are shown as perpendicular labels on the x-axis, but one can choose to display specific model names only (e.g., `whichticks=c("model13","model20")`). The tick mark labels can be made horizontal, by putting parameter `las` to 0.

Solid lines (only shown in case of 20 or less samples) indicate the scree test values per sample, and symbols indicate the top three of the models per sample. The symbols can be adjusted with the parameter `pch` and the colors with `col`. The model (or multiple models) that is selected most often across samples, is indicated with a horizontal line.

See Also

`CHull`

Examples

```
data <- cbind(c(305,456,460,607,612,615,758,764,768,770,909,916,921,924),
c(152,89,79,71,57,57,64,49,47,47,60,41,39,39))
test <- array(rnorm(14*20,sd=2.5),c(14,20))
for (i in 1:20){
  data <- cbind(data,data[,2]+test[,i])
}

output <- MultiCHull(data)
summary(output)
plot(output)
```

`runShinyApp` *Run Shiny App "multichull"*

Description

Open the graphical user-interface for applying the (multi)CHull procedure

Usage

```
runShinyApp()
```

Index