

Package ‘MOCCA’

May 7, 2026

Title Multi-Objective Optimization for Collecting Cluster Alternatives

Version 1.4

Date 2020-03-11

Author Johann Kraus <johann.kraus@uni-ulm.de>

Maintainer Hans Kestler <hans.kestler@uni-ulm.de>

Description

Provides methods to analyze cluster alternatives based on multi-objective optimization of cluster validation indices. For details see Kraus et al. (2011) <[doi:10.1007/s00180-011-0244-6](https://doi.org/10.1007/s00180-011-0244-6)>.

Depends R (>= 2.0.0), cclust, clue, cluster, class

License Artistic License 2.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-17 10:20:02 UTC

Contents

MOCCA-package	1
analyzePareto	2
mocca	3
toy5	4
toy9	5
Index	6

MOCCA-package	<i>Multi-objective optimization for collecting cluster alternatives</i>
---------------	---

Description

This package provides methods to analyze cluster alternatives based on multi-objective optimization of cluster validation indices.

Details

Package: MOCCA
Version: 1.4
Date: 2020-03-05
Depends: R (>= 2.0.0), cclust, clue, cluster, class
License: Artistic License 2.0

Estimating the optimal cluster number of a dataset is often a difficult problem. Cluster validation indices are designed to rate a clustering and can be used to rank different cluster sizes. Bootstrapping has been proposed to determine robust cluster numbers based on such indices. However, these estimations vary depending on the employed clustering algorithm and cluster validation index. The idea of MOCCA is to estimate robust cluster numbers by aggregating the best cluster numbers of several clustering algorithms and cluster validation indices in a multi-objective setting.

The main function of the package is `mocca`, which applies multiple cluster algorithms to a cluster dataset in a bootstrapping setting and calculates several cluster validation indices. These results can be compared by calculating the Pareto-optimal cluster sizes and ranking them according to their domination. This is implemented in `analyzePareto`.

Author(s)

Johann Kraus <johann.kraus@uni-ulm.de> Maintainer: Hans Kestler <hans.kestler@uni-ulm.de>

Examples

```
data(toy5)
obj <- mocca(toy5, R=10, K=2:5)
print(analyzePareto(obj$objectiveVals))
```

analyzePareto

Analyze the Pareto-optimal cluster sizes

Description

Computes the set of Pareto-optimal cluster sizes in `obj` according to the values of the cluster validation indices. A ranking of optimal cluster sizes and a table illustrating the ranking of solutions are returned.

Usage

```
analyzePareto(obj)
```

Arguments

`obj` A matrix returned by `mocca` in its `objectiveVals` component. This matrix contains the values of several cluster validation indices for different cluster algorithms and different cluster sizes.

Value

A list with the following components

rank	A vector containing the ranking of the Pareto-optimal cluster sizes.
table	A table specifying the ranking of Pareto-optimal cluster sizes. Each row is associated with a particular Pareto-optimal cluster size. Its entries specify in how many objective functions it dominates clusterings of other cluster sizes. The Pareto-optimal cluster sizes are ranked by the minimum number of objectives in which they dominate other cluster sizes.

Examples

```
set.seed(12345)
data(toy5)
obj <- mocca(toy5, R=10, K=2:5)
print(analyzePareto(obj$objectiveVals))
```

 mocca

Multi-objective optimization for collecting cluster alternatives

Description

Performs a multi-objective optimization for collecting cluster alternatives. The algorithm draws R bootstrap samples from x . It calculates clusterings for all specified cluster numbers K using k-means, neuralgas, and single-linkage clustering. It then applies several cluster validation indices to the clusterings.

Usage

```
mocca(x, R = 50, K = 2:10, iter.max = 1000, nstart = 10)
```

Arguments

<code>x</code>	A numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with numeric columns).
<code>R</code>	The number of bootstrap samples.
<code>K</code>	The range of cluster numbers, i.e. a vector of integers listing the maximum numbers of clusters to be used by each of the algorithms.
<code>iter.max</code>	The maximum number of iterations allowed in k-means.
<code>nstart</code>	For k-means, how many random sets should be chosen?

Value

A list with two entries:

- `cluster` A list containing one sublist for each clustering algorithm and the baseline cluster solution. Each of these lists hold an entry for each cluster size K , which again consists of R vectors of cluster assignments. These vectors assign each data point in x to a cluster.
- `objectiveVals` A matrix of objective function values. Each row corresponds to a certain cluster validation index applied to a certain clustering algorithm. The columns correspond to different cluster numbers. Consequently, an entry of the matrix specifies the median value of a certain cluster validation index for a certain clustering algorithm with a specific number of clusters over the R bootstrap samples.

Examples

```
set.seed(12345)
data(toy5)
res <- mocca(toy5, R=10, K=2:5)
print(res$objectiveVals)
# plot kmeans result for MCA index against neuralgas result for MCA index
plot(res$objectiveVals[1,], res$objectiveVals[5,], pch=NA,
      xlab=rownames(res$objectiveVals)[1], ylab=rownames(res$objectiveVals)[5])
text(res$objectiveVals[1,], res$objectiveVals[5,], labels=colnames(res$objectiveVals))
```

toy5

Toy data set with 5 clusters

Description

This artificial data set contains 5 two-dimensional Gaussian clusters.

Usage

```
data(toy5)
```

Format

`toy5` is a matrix with 50 cases (rows) and 2 variables (columns).

Examples

```
data(toy5)
plot(toy5)
```

`toy9`*Toy data set with 9 clusters*

Description

This artificial data set contains 9 two-dimensional Gaussian clusters.

Usage

```
data(toy5)
```

Format

`toy9` is a matrix with 90 cases (rows) and 2 variables (columns).

Examples

```
data(toy9)  
plot(toy9)
```

Index

* **datasets**

toy5, [4](#)

toy9, [5](#)

* **package**

MOCCA-package, [1](#)

analyzePareto, [2](#), [2](#)

MOCCA (MOCCA-package), [1](#)

mocca, [2](#), [3](#)

MOCCA-package, [1](#)

toy5, [4](#)

toy9, [5](#)