

Package: oclust (via r-universe)

May 10, 2026

Title Gaussian Model-Based Clustering with Outliers

Date 2025-07-07

Version 1.0.0

Imports entropy, graphics, stats, utils, mclust, mixture, dbscan,
MASS, mvtnorm, progress

Suggests testthat

Description Provides a function to detect and trim outliers in
Gaussian mixture model-based clustering using methods described
in Clark and McNicholas (2024)
<[doi:10.1007/s00357-024-09473-3](https://doi.org/10.1007/s00357-024-09473-3)>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Katharine M. Clark [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-6162-2300>>), Paul D. McNicholas
[aut] (ORCID: <<https://orcid.org/0000-0002-2482-523X>>)

Maintainer Katharine M. Clark <katclark@trentu.ca>

Config/pak/sysreqs libgsl0-dev

Repository <https://clarkkm2.r-universe.dev>

Date/Publication 2025-07-08 02:24:54 UTC

RemoteUrl <https://github.com/cran/oclust>

RemoteRef HEAD

RemoteSha 5c9bffcfe78e82456a1a0bff14d3183dec3d75fd

Contents

findGrossOuts	2
minMD	3
MixBetaDens	3

oclust	4
plot.oclust	6
print.oclust	7
print.summary.oclust	8
simOuts	8
summary.oclust	9

Index	10
--------------	-----------

findGrossOuts	<i>Find Initial Gross Outliers</i>
---------------	------------------------------------

Description

findGrossOuts uses DBSCAN to find areas of high density. Mahalanobis distance to the closest area of high density is calculated for each point. With no elbow specified, the Mahalanobis distances are plotted. If the elbow is specified, the indices of the gross outliers are returned.

Usage

```
findGrossOuts(X, minPts = 10, xlim = NULL, elbow = NULL)
```

Arguments

X	A data matrix
minPts	The minimum number of points in each region of high density. Default is 10
xlim	A vector of form c(xmin,xmax) to specify the domain of the plot. Default is NULL, which sets xmax to 10% of the data size.
elbow	An integer specifying the location of the elbow in the plot of Mahalanobis distances. Default is NULL, which returns the plot. If elbow is specified, no plot is produced and the gross outliers are returned.

Details

The function plots Mahalanobis distance to the closest centre in decreasing order or returns the indices of the gross outliers. The elbow location of the plot provides a good indication as to where the gross outliers end. Running the function first without an elbow specified will plot the Mahalanobis distances. Running it again with the elbow specified will return the outliers. It is recommended to choose the elbow conservatively. If the MDs decrease smoothly, there are no gross outliers. Set elbow=1.

Value

findGrossOuts returns a vector with the indices of the gross outliers. One fewer point is returned than the elbow specified.

minMD	<i>Minimum Mahalanobis Distance</i>
-------	-------------------------------------

Description

minMD calculates the Mahalanobis distance to each cluster and returns the Mahalanobis distance to the closest cluster.

Usage

```
minMD(X, sigs, mus)
```

Arguments

X	A matrix or data frame of the data.
sigs	A list of cluster variance matrices
mus	A list of cluster mean vectors

Details

This function is used to help identify initial gross outliers.

Value

minMD returns a vector of length n corresponding to the minimum MD for each point.

MixBetaDens	<i>Mixture of Beta Functions</i>
-------------	----------------------------------

Description

MixBetaDens generates the pdf and cdf of a mixture of beta functions, and calculates the area under the graph between two points.

Usage

```
MixBetaDens(  
  n,  
  p,  
  x = seq(0, 15, by = 0.01),  
  a = 0,  
  b = 1,  
  n_g = n_g,  
  var = var  
)
```

Arguments

n	The number of observations in the dataset
p	The dimension
x	A vector of x values to evaluate. Default value is seq(0, 15, by=0.01)
a	Lower bound for area evaluation. Default value is 0
b	Upper bound for area evaluation. Default value is 1
n_g	Vector describing the number of observations in each cluster
var	A list of cluster variance matrices

Details

The domain for this function is not [0,1] as is typical with a beta function. The domain encompasses the shifted log-likelihoods generated in [oclust](#).

Value

MixBetaDens returns a list with

pdf	The probability density at each x value
cdf	The cumulative density at each x value
area	The area under the pdf graph between a and b

 oclust

The OCLUST Algorithm

Description

oclust is a trimming method in model-based clustering. It iterates over possible values for the number of outliers and returns the model parameters for the best model as determined by the minimum KL divergence. If kuiper=TRUE, oclust calculates an approximate p-value using the Kuiper test and stops the algorithm if the p-value exceeds the specified threshold.

Usage

```
oclust(
  X,
  max0,
  G,
  grossOuts = NULL,
  modelNames = "VVV",
  mc.cores = 1,
  nmax = 1000,
  kuiper = FALSE,
  pval = 0.05,
  B = 100,
```

```

    verb = FALSE,
    scale = TRUE
  )

```

Arguments

<code>X</code>	A matrix or data frame with <code>n</code> rows of observations and <code>p</code> columns
<code>max0</code>	An upper bound for the number of outliers
<code>G</code>	The number of clusters
<code>grossOuts</code>	The indices of the initial outliers to remove. Default is <code>NULL</code> .
<code>modelNames</code>	The model to fit using the <code>gpcm</code> function in the <code>mixture</code> package. Default is <code>"VVV"</code> (unconstrained). If <code>modelNames=NULL</code> , all models are fitted for each subset at each iteration. The BIC chooses the best model for each subset.
<code>mc.cores</code>	Number of cores to use if running in parallel. Default is <code>1</code>
<code>nmax</code>	Maximum number of iterations for each EM algorithm. Decreasing <code>nmax</code> may speed up the algorithm but lose precision in finding the log-likelihoods.
<code>kuiper</code>	A logical specifying whether to use the Kuiper test (Kuiper, 1960) to stop the algorithm when p-value exceeds the specified threshold. Default is <code>FALSE</code> .
<code>pval</code>	The p-value for the Kuiper test. Default is <code>0.05</code> .
<code>B</code>	Number of samples to calculate the approximate p-value. Default is <code>100</code> .
<code>verb</code>	A logical specifying whether to print the current iteration number. Default is <code>FALSE</code>
<code>scale</code>	A logical specifying whether to centre and scale the data. Default is <code>TRUE</code>

Details

Gross outlier indices can be found with the [findGrossOuts](#) function.

N. H. Kuiper, Tests concerning random points on a circle, in: *Nederl. Akad. Wetensch. Proc. Ser. A*, Vol. 63, 1960, pp. 38–47.

Value

`oclust` returns a list of class `oclust` with

<code>data</code>	A list containing the raw and scaled data
<code>num0</code>	The predicted number of outliers
<code>outliers</code>	The most likely outliers in the optimal solution in order of likelihood
<code>class</code>	The classification for the optimal solution
<code>model</code>	The model selected for the optimal solution
<code>G</code>	The number of clusters
<code>pi.g</code>	The group proportions for the optimal solution
<code>mu</code>	The cluster means for the optimal solution
<code>sigma</code>	The cluster variances for the optimal solution
<code>KL</code>	The KL divergence for each iteration, with the first value being for the initial dataset with the gross outliers removed
<code>allCand</code>	All outlier candidates in order of likelihood

Examples

```

# simulate 4D dataset
library(mvtnorm)
set.seed(123)
data <- rbind(rmvnorm(250, rep(-3, 4), diag(4)),
             rmvnorm(250, rep(3, 4), diag(4)))
# add outliers
noisy <- simOuts(data = data, alpha = 0.02, seed = 123)

# Find gross outliers
findGrossOuts(X = noisy, minPts = 10)

# Elbow between 5 and 10. Specify limits of graph
findGrossOuts(X = noisy, minPts = 10, xlim = c(5, 10))

# Elbow at 9
gross <- findGrossOuts(X = noisy, minPts = 10, elbow = 9)

# run algorithm
if (interactive()) {
# This example takes a few minutes to run
  result <- oclust(X = noisy, max0 = 15, G = 2, grossOuts = gross,
                  modelNames = "EEE", mc.cores = 1, nmax = 50,
                  kuiper = FALSE, verb = TRUE, scale = TRUE)
}

```

plot.oclust

Plots results of the 'oclust' algorithm.

Description

Plots results of the 'oclust' algorithm.

Usage

```

## S3 method for class 'oclust'
plot(
  x,
  what = c("classification", "KL", "pval"),
  dimens = NULL,
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  addEllipses = TRUE,
  ...
)

```

Arguments

x	An 'oclust' class object obtained by using <code>oclust</code>
what	A string specifying the type of graph. The options are: "classification" a plot of the classifications for the optimal solution. For data with $p > 2$, if more than two "dimens" are specified, a pairs plot is produced. If two "dimens" are specified, a coordinate projection plot is produced with the specified "dimens". Ellipses corresponding to covariances of mixture components are also drawn if "addEllipses = TRUE". "KL" a plot of Kullback-Leibler divergence for each number of outliers. "pval" a plot of approximate p-value for each number of outliers.
dimens	a vector specifying the dimensions of the coordinate projections
xlab, ylab	optional argument specifying axis labels for the classification plot
ylim	optional limits of the y axis of the BIC and KL plots
addEllipses	logical indicating whether to include ellipses corresponding to the covariances of the mixture components
...	other graphical parameters

print.oclust

Print oclust

Description

Prints list of available components for 'oclust' class objects.

Usage

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

Arguments

x	An 'oclust' class object obtained by using <code>oclust</code>
...	additional print parameters

```
print.summary.oclust Prints the summary of key results for 'oclust' class objects.
```

Description

Prints the summary of key results for 'oclust' class objects.

Usage

```
## S3 method for class 'summary.oclust'
print(x, digits = getOption("digits"), ...)
```

Arguments

x	An 'oclust' class object obtained by using <code>oclust</code>
digits	number of digits to print
...	additional print arguments

```
simOuts Simulate Outliers
```

Description

simOuts generates uniform outliers in each dimension in (min- 2.range, max+ 2.range)

Usage

```
simOuts(data, alpha, seed = 123)
```

Arguments

data	The data in data frame form
alpha	The proportion of outliers to add in terms of the original data size
seed	Set the seed for reproducibility

Value

simOuts returns a data frame with the generated outliers appended to the original data.

summary.oclust	<i>Summarizes key results for 'oclust' class objects.</i>
----------------	---

Description

Summarizes key results for 'oclust' class objects.

Usage

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

Arguments

object	An 'oclust' class object obtained by using oclust
...	additional summary arguments

Index

`findGrossOuts`, [2](#), [5](#)

`minMD`, [3](#)

`MixBetaDens`, [3](#)

`oclust`, [4](#), [4](#), [7–9](#)

`plot.oclust`, [6](#)

`print.oclust`, [7](#)

`print.summary.oclust`, [8](#)

`simOuts`, [8](#)

`summary.oclust`, [9](#)