

# Package ‘RMBC’

January 20, 2025

**Type** Package

**Title** Robust Model Based Clustering

**Version** 0.1.0

**Author** Juan Domingo Gonzalez [cre, aut],  
Victor J. Yohai [aut],  
Ruben H. Zamar [aut],  
Ricardo Maronna [aut]

**Maintainer** Juan Domingo Gonzalez <juanrst@hotmail.com>

**Description** A robust clustering algorithm (Model-Based) similar to Expectation Maximization for finite mixture normal distributions is implemented, its main advantage is that the estimator is resistant to outliers, that means that results of parameter estimation are still correct when there are atypical values in the sample (see Gonzalez, Maronna, Yohai and Zamar (2021) <[arxiv:2102.06851](https://arxiv.org/abs/2102.06851)>).

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends** R (>= 3.5.0), stats

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** tclust, knitr, testthat (>= 2.1.0), rmarkdown

**Imports** ktaucenters, mvtnorm, MASS

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-07-22 06:40:05 UTC

## Contents

is_in_gr . . . . .	2
klfor2normals . . . . .	3
phytoplankton_acoustic_data . . . . .	3

quad_disc . . . . .	5
RMBC . . . . .	5
RMBCaux . . . . .	7
robustINIT . . . . .	8
sumkl . . . . .	9
weightedMscale . . . . .	10
weightedSestimator . . . . .	10
weightW . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

<i>is_in_gr</i>	<i>is_in_gr</i>
-----------------	-----------------

---

## Description

Given Y data and a set of mixture parameters, this function returns a boolean vector B whose length is equal than Y length. A[i] is TRUE if only if Y[i] does not belong to the union of confidence ellipsoids of level given by the cutoff parameter.

## Usage

```
is_in_gr(Y, cutoff = 0.999, theta.mu, theta.sigma)
```

## Arguments

Y	A matrix of size n x p.
cutoff	quantiles of chi-square to be used as a threshold for outliers detection, defaults to 0.999
theta.mu	The estimated centers: A list with K elements, each of them is an array of length p.
theta.sigma	The estimated scatter matrices: A list with K matrices, each of them has dimension p x p

## Value

A boolean vector of length n, true at j-th location indicates that the j-th element of Y is a regular observation (that is, it is not considered as an outlier)

---

klfor2normals	<i>klfor2normals Compute the Kullback-Leibler divergence for 2 normal multivariate distributions</i>
---------------	--

---

### Description

klfor2normals Compute the Kullback-Leibler divergence for 2 normal multivariate distributions

### Usage

```
klfor2normals(theta1.mu, theta1.sigma, theta2.mu, theta2.sigma)
```

### Arguments

theta1.mu	the location parameter of the first distribution
theta1.sigma	the covariance matrix of the first distribution
theta2.mu	the location parameter of the second distribution
theta2.sigma	the covariance matrix of the second distribution

### Value

the K-L divergence.

---

phytoplankton_acoustic_data	<i>Phytoplankton_acoustic_data</i>
-----------------------------	------------------------------------

---

### Description

Data obtained by taking laboratory measurements of ultrasonic acoustic signals: a pulse is emitted by a transducer, this pulse interacts with phytoplankton suspended in the water and produces an acoustic dispersion (scattering), which is recorded by an electronic acquisition device. A filtering process of the signal is performed in a first stage. Portions of the signal belong to one of the two main cases:

- (a) Signals corresponding to the acoustic response of phytoplankton
- (b) Signals corresponding to spurious dispersers, such as bubbles or particles in suspension, whose intensity is greater than in case (a).

To classify a signal in one of these two groups biologists create a vector (X1, X2) defined as follows:

- X1 = ratio of filtered to non-filtered signal power
- X2 = filtered signal power expressed in dB.

The available data consists of 375 such measurements. These data is particularly useful to compare robust procedures because 20 to be outliers produced by a communication failure between the electronic device (digital oscilloscope) and the software for acquiring the acoustic signal. This failure occurs once every 5 microseconds, which allows the scientists to identify the outliers. The outliers appear as a separated group in the region  $X1 < 0.5$  and  $X2 > 20$ .

**Usage**

```
phytoplankton_acoustic_data
```

**Format**

a list of length 2, where its elements are

- Y: A matrix of dimension 375 x 2, each row contains X1 and X2 values
- outliers\_index: An array with the outliers index-locations

**References**

- [1] Cinquini, M., Bos, P., Prario, I and Blanc, S. (2016), “Advances on modelling, simulation and signal processing of ultrasonic scattering responses from phytoplankton cultures,” in Proceedings of Meetings on Acoustics 22ICA, 28, American Society of Acoustics.
- [2] Gonzalez J.D, Maronna R., Yohai V., & and Zamar . (2021). Robust Model-Based Clustering. arXiv preprint <<https://arxiv.org/abs/2102.06851>>

**Examples**

```
#####
# upload matrix #####
#####

Y <- phytoplankton_acoustic_data$Y

outliers_index <- phytoplankton_acoustic_data$outliers_index

Yclean=Y[-outliers_index,]

trueOutliers=Y[outliers_index,]

#####
# plot results #####
#####

plot(Y, main = "Phytoplankton acoustic data", cex.main = 3, lwd = 1,pch = 19, cex = 1,
      type = "n", xlab = "x1", ylab = "x2", xlim = c(0,1.1), ylim = c(0,43)
      )

points(trueOutliers,lwd=2,cex=1,pch=4)

points(Yclean,col=1,lwd=1.5,pch=21, bg=4, cex=1)
```

---

quad_disc	<i>quad_disc</i>
-----------	------------------

---

**Description**

Computes the quadratic discriminant of each mixture component,

**Usage**

```
quad_disc(Y, theta.alpha, theta.mu, theta.sigma)
```

**Arguments**

Y	A matrix of size n x p.
theta.alpha	The alpha values: An array of K positive real numbers they must verify the condition $\sum(\text{thetaOld.mu}) = 1$ .
theta.mu	The estimated centers: A list with K elements, each of them is an array of length p.
theta.sigma	The estimated scatter matrices: A list with K matrices, each of them has dimension p x p

**Value**

A n x K matrix, where each row has the values of the quadratic discriminant with regarding to the j-th mixture component,  $j = 1, \dots, K$

---

RMBC	<i>Robust Model Base Clustering a robust and efficient version of EM algorithm.</i>
------	---

---

**Description**

Robust Model Base Clustering a robust and efficient version of EM algorithm.

**Usage**

```
RMBC(Y, K, max_iter = 80, tolerance = 1e-04)
```

**Arguments**

Y	A matrix of size n x p.
K	The number of clusters.
max_iter	a maximum number of iterations used for the algorithm stopping rule
tolerance	tolerance parameter used for the algorithm stopping rule

**Value**

A list including the estimated mixture distribution parameters and cluster-label for the observations

- alpha: K numeric values representing the convex combination coefficients.
- mu: a list of length K with the location initial estimators.
- sigma: a list of length K with the location scatter matrix estimators.
- nonoutliers: an array of indices that contains the estimated nonoutliers observations
- outliers: an array of indices that contains the estimated outliers observations
- cluster: A vector of integers (from 0:k) indicating the cluster to which each point is allocated. 0 label corresponds to outliers
- cluster\_without\_outliers: A vector of integers (from 1:k) indicating the cluster to which each point is allocated. Note: Outliers are assigned to the cluster that maximizes the belonging probability. To recognize the outliers the field outliers should be used

**Examples**

```
# Generate Sintetic data (three normal cluster in two dimension)
# clusters have different shapes and orentation.
# The data is contaminated uniformly (level 20%).
#####
#### Start data generating process #####
#####

# generates base clusters

Z1 <- c(rnorm(50,0),rnorm(50,0),rnorm(50,0))
Z2 <- rnorm(150);
X <- matrix(0, ncol=2,nrow=150);
X[,1]=Z1;X[,2]=Z2
true.cluster= c(rep(1,50),rep(2,50),rep(3,50))
# rotate, expand and translate base clusters
theta=pi/3;
aux1=matrix(c(cos(theta),-sin(theta),sin(theta),cos(theta)),nrow=2)

aux2=sqrt(4)*diag(c(1,1/4))

B=aux1%*%aux2%*%t(aux1)

X[true.cluster==3,]=X[true.cluster==3,]%*%aux2%*%aux1 +
matrix(c(15,2), byrow = TRUE,nrow=50,ncol=2)
X[true.cluster==2,2] = X[true.cluster==2,2]*4
X[true.cluster==1,2] = X[true.cluster==1,2]*0.1
X[true.cluster==1, ] = X[true.cluster==1,]+
matrix(c(-15,-1),byrow = TRUE,nrow=50,ncol=2)

### Generate 30 sintetic outliers (contamination level 20%)

outliers=sample(1:150,30)
X[outliers, ] <- matrix(runif( 60, 2 * min(X), 2 * max(X) ),
                      ncol = 2, nrow = 30)
```

```
#####
#### END data generating process #####
#####

### APPLYING RMBC ALGORITHM

ret = RMBC(Y=X, K=3,max_iter = 82)

cluster = ret$cluster
#####
### plotting results #####
#####
oldpar=par(mfrow=c(1,2))
plot(X, main="actual clusters" )
for (j in 1:3){
  points(X[true.cluster==j,],pch=19, col=j+1)
}
points(X[outliers,],pch=19,col=1)

plot(X,main="clusters estimation")
for (j in 1:3){
  points(X[cluster==j,],pch=19, col=j+1)
}
points(X[ret$outliers,],pch=19,col=1)
par(oldpar)
```

---

RMBCaux

*RMBCaux*


---

## Description

Robust Model Base Clustering algorithm based on centers, a robust and efficient version of EM algorithm.

## Usage

```
RMBCaux(
  Y,
  K,
  thetaOld.alpha,
  thetaOld.mu,
  thetaOld.sigma,
  max_iter,
  niterFixedPoint,
  tolerance,
  cutoff = 1 - 0.001
)
```

**Arguments**

Y	A matrix of size $n \times p$ .
K	The number of clusters.
thetaOld.alpha	The initial alpha: An array of K positive real numbers they must verify the condition $\sum(\text{thetaOld.mu}) = 1$ .
thetaOld.mu	The initial centers: A list with K elements, each of them is an array of length p.
thetaOld.sigma	The initial scatter matrix: A list with K matrix, each of them has dimension $p \times p$
max_iter	a maximum number of iterations used for the algorithm stopping rule
niterFixedPoint	the maximum number of iteration in the internal loop which computes sigma and mu separately. The default value is niterFixedPoint=1
tolerance	tolerance parameter used for the algorithm stopping rule
cutoff	optional argument for outliers detection - quantiles of chi-square to be used as a threshold for outliers detection, defaults to 0.999

**Value**

A list including the estimated K centers and labels for the observations

- centers: matrix of size  $K \times p$ , with the estimated K centers.
- cluster: array of size  $n \times 1$  integers labels between 1 and K.
- tauPath: sequence of tau scale values at each iterations.
- Wni: numeric array of size  $n \times 1$  indicating the weights associated to each observation.
- emptyClusterFlag: a boolean value. True means that in some iteration there were clusters totally empty
- niter: number of iterations until convergence is achieved or maximum number of iteration is reached
- didistance of each observation to its assigned cluster-center

---

robustINIT

*robustINIT*


---

**Description**

Robust Initializer for RMBC algorithm, it depends on the package ktaucenters

**Usage**

```
robustINIT(Y, K, nstart = 10)
```



**Arguments**

Y	A matrix of size $n \times p$ .
K	The number of groups
nstart	the number of starting points to the algorithm, defaults to 10

**Value**

A list including the initial parameters of the mixture distribution, namely

- `alphaINIT`: K numeric values representing the convex combination coefficients.
- `muINIT`: a list of length K with the location initial estimators.
- `sigmaINIT`: a list of length K with the scatter matrix estimators.
- `indicesINIT`: indices with initial clusters

---

sumkl	<i>sumkl</i> The sum of K-L divergence measure between two successive iterations for each component of a mixture distribution,
-------	--

---

**Description**

sumkl The sum of K-L divergence measure between two successive iterations for each component of a mixture distribution,

**Usage**

```
sumkl(thetaNew.mu, thetaNew.sigma, thetaOld.mu, thetaOld.sigma)
```

**Arguments**

<code>thetaNew.mu</code>	the location parameters of the first distribution
<code>thetaNew.sigma</code>	the covariance matrix of the first distribution
<code>thetaOld.mu</code>	the location parameter of the second distribution
<code>thetaOld.sigma</code>	the covariance matrix of the second distribution

**Value**

the K-L divergence.

---

weightedMscale	<i>weightedMscale the M scale of an univariate sample (see reference below)</i>
----------------	---

---

### Description

weightedMscale the M scale of an univariate sample (see reference below)

### Usage

```
weightedMscale(u, b = 0.5, weights, c, initialsc = 0)
```

### Arguments

u	an univariate sample of size n.
b	the desired break down point
weights	the weights of each observation.
c	a tuning constant, if consistency to standard normal distribution is desired use <a href="#">normal_consistency_constants</a>
initialsc	the initial scale value, defaults to 0

### Value

the weighted-Mscale value

### References

Maronna, R. A., Martin, R. D., Yohai, V. J., & Salibián-Barrera, M. (2018). Robust statistics: theory and methods (with R). Wiley.

---

weightedSestimator	<i>weightedSestimator</i>
--------------------	---------------------------

---

### Description

Computes the weighted location and scatter matrix estimators of the j-th mixture component , where the weights are calculated in the expectation-step.

**Usage**

```

weightedSestimator(
  Y,
  mu_init,
  sigma_init,
  max_iterFP = 1,
  weights,
  fixed_alpha
)

```

**Arguments**

Y	A matrix of size n x p.
mu_init	The previously computed center: an numerical array of length p.
sigma_init	The previously computed scatter matrix: an array of numeric values p x p
max_iterFP	the maximum number of fixed point iterations used for the algorithm, defaults to 1
weights	The weights that contain the probability membership of each observation (related to the overall mixture components)
fixed_alpha	the fixed alpha value for the corresponding mixture component

**Value**

A list including the estimated K centers and labels for the observations  $\text{list}(\text{cov}=\text{matrixSigma}, \text{covAux1}=\text{covAux1}, \text{mu}=\text{muk}, \text{s}=\text{s})$

- cov: the computed weighted scatter matrix
- mu: the computed weighted center
- s: the weighted scale factor s.

---

weightW

*weightW*


---

**Description**

Weight function ktaucenters

**Usage**

```
weightW(arg, p)
```

**Arguments**

arg	An 1-D array containing the distances.
p	the dimension of the element

**Value**

an array of the same size of arg with the value of the weights

# Index

## \* datasets

- phytoplankton\_acoustic\_data, [3](#)
- is\_in\_gr, [2](#)
- klfor2normals, [3](#)
- normal\_consistency\_constants, [10](#)
- phytoplankton\_acoustic\_data, [3](#)
- quad\_disc, [5](#)
- RMBC, [5](#)
- RMBCaux, [7](#)
- robustINIT, [8](#)
- sumkl, [9](#)
- weightedMscale, [10](#)
- weightedSestimator, [10](#)
- weightW, [11](#)