# Package 'RGMM'

November 24, 2023

**Type** Package

**Title** Robust Mixture Model

**Version** 2.1.0

**Description** Algorithms for estimating robustly the parameters of a Gaussian, Student, or Laplace Mixture Model.

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** Rcpp, foreach, doParallel, mvtnorm,mclust,parallel,LaplacesDemon, genieclust, RSpectra, ggplot2, reshape2, DescTools

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.1.2

**NeedsCompilation** yes

**Author** Antoine Godichon-Baggioni [aut, cre, cph], Stéphane Robin [aut]

**Maintainer** Antoine Godichon-Baggioni <antoine.godichon_baggioni@upmc.fr>

**Repository** CRAN

**Date/Publication** 2023-11-24 09:20:07 UTC

## R topics documented:

RGMM-package *Robust Mixture Model*

## Description

In this package, we provide functions to provide robust clustering in the case of Gaussian, Student and Laplace Mixture Models. Function RobVar computes robustly the covariance of a numerical data set which are realizations of Gaussian, Student or Laplace vectors. Function RobMM enables to provide a clustering of a numerical data set, RMMplot enables to produce graph for Robust Mixture Models, while Gen_MM enables to generate possibly contaminated mixture of Gaussian, Student and Laplace vectors.

## Author(s)

NA

Maintainer: NA

## References

Cardot, H., Cenac, P. and Zitt, P-A. (2013). Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19, 18-43.

Cardot, H. and Godichon-Baggioni, A. (2017). Fast Estimation of the Median Covariation Matrix with Application to Online Robust Principal Components Analysis. *Test*, 26(3), 461-480

Vardi, Y. and Zhang, C.-H. (2000). The multivariate L1-median and associated data depth. *Proc. Natl. Acad. Sci. USA*, 97(4):1423-1426.

Gen_MM *Gen_MM*

## Description

Generate a sample of a Mixture Model

## Usage

```
Gen_MM(nk=NA, df=3, mu=NA, Sigma=FALSE, delta=0,cont="Student",
                model="Gaussian", dfcont=1, mucont=FALSE, Sigmacont=FALSE,
                minU=-20, maxU=20)
```

## Arguments

| | |
|---|---|
| nk | An integer vector containing the desired number of data for each class. The defulat is nk=rep(500,3). |
| df | An integer larger (or qual) than 3 specifying the degrees of freedom of the Student law, if model='Student'. Default is 3. |
| mu | A numeric matrix whose raws correspond to the centers of the classes. By default, mu is generated randomly. |
| Sigma | An array containing the variance of each class. See exemple for more details. |
| delta | A positive scalr between 0 and 1 giving the proportion of contaminated data. Default is 0. |
| cont | The kind of contamination chosen. Can be equal to 'Unif' or 'Student'. |
| model | A string character specifying the model chosen for the Mixture Model. Can be equal to 'Gaussian' (default) or 'Student'. |
| dfcont | A positive integer specifying the degrees of freedom of the contamination laws if cont='Student'. Default is 1. |
| mucont | A numeric matrix whose rows correspond to the centers of the contamination laws. By default, mucont is chosen equal to mu. |
| Sigmacont | An array containing the variance of each contamination law. By default, sigmacont is chosen equal to sigma. |
| minU | A scalar giving the lower bound of the uniform law of the contamination if cont='Unif'. |
| maxU | A scalar giving the upper bound of the uniform law of the contamination if cont='Unif'. |

## Value

A list with:

| | |
|---|---|
| Z | An integer vector specifying the true classification. If delta is non nul, the contaminated data are consider as a class. |
| C | A 0-1 vector specifying the contaminated data. |
| X | A numerical matrix giving the generated data. |

## See Also

See also RobMM and RobVar.

## Examples

```
p <- 3
nk <- rep(50,p)
mu <- c()
for (i in 1:length(nk))
{
  Z <- rnorm(3)
```

```
  mu <- rbind(mu,length(nk)*Z/sqrt(sum(Z^2)))
}
Sigma <- array(dim=c(length(nk), p, p))
for (i in 1:length(nk))
{
  Sigma[i, ,] <- diag(p)
}
ech <- Gen_MM(nk=nk,mu=mu,Sigma=Sigma)
```

---

RMMplot *RMMplot*

---

## Description

A plot function for Robust Mixture Model

## Usage

```
RMMplot(a,outliers=TRUE,
    graph=c('Two_Dim','Two_Dim_Uncertainty','ICL','BIC',
    'Profiles','Uncertainty'),bestresult=TRUE,K=FALSE)
```

## Arguments

| | |
|---|---|
| a | Output from RobMM. |
| outliers | An argument telling if there are outliers or note. In this case, Two dimensional plots and profiles plots will be done without detected outliers. Default is TRUE. |
| graph | A string specifying the type of graph requested. Default is c('Two_Dim','Two_Dim_Uncertainty','ICL 'Profiles','Uncertainty'). |
| bestresult | A logical indicating if the graphs must be done for the result chosen by the selected criterion. Default is TRUE. |
| K | A logical or positive integer giving the chosen number of clusters for each the graphs should be drawn. |

## See Also

See also RobMM and Gen_MM.

## Examples

```
## Not run:
ech <- Gen_MM(mu = matrix(c(rep(-2,3),rep(2,3),rep(0,3)),byrow = TRUE,nrow=3))
 X <- ech$X
 res <- RobMM(X , nclust=3)
 RMMplot(res,graph=c('Two_Dim'))

## End(Not run)
```

RobMM *RobMM*

## Description

Robust Mixture Model

## Usage

```
RobMM(X, nclust=2:5, model="Gaussian", ninit=10,
               nitermax=50, niterEM=50, niterMC=50, df=3,
               epsvp=10^(-4), mc_sample_size=1000, LogLike=-Inf,
               init='genie', epsPi=10^-4, epsout=-20,scale='none',
               alpha=0.75, c=ncol(X), w=2, epsilon=10^(-8),
               criterion='BIC',methodMC="RobbinsMC", par=TRUE,
               methodMCM="Weiszfeld")
```

## Arguments

| | |
|---|---|
| X | A matrix giving the data. |
| nclust | A vector of positive integers giving the possible number of clusters. |
| model | The mixture model. Can be 'Gaussian' (by default), 'Student' and 'Laplace'. |
| ninit | The number of random initisalizations. Befault is 10. |
| nitermax | The number of iterations for the Weiszfeld algorithm if MethodMCM= 'Weiszfeld'. |
| niterEM | The number of iterations for the EM algorithm. |
| niterMC | The number of iterations for estimating robustly the variance of each class if methodMC='FixMC' or methodMC='GradMC'. |
| df | The degrees of freedom for the Student law if model='Student'. |
| scale | Run the algorithm on scaled data if scale='robust'. |
| epsvp | The minimum values the estimates of the eigenvalues of the Median Covariation Matrix can take. Default is 10^-4. |
| mc_sample_size | The number of data generated for the Monte-Carlo method for estimating robustly the variance. |
| LogLike | The initial loglikelihood to "beat". Defulat is -Inf. |
| init | Can be F if no non random initialization of the algorithm is done, 'genie' if the algorithm is initialized with the help of the function 'genie' of the package genieclust or 'Mclust' if the initialization is done with the function hclass of the package Mclust. |
| epsPi | A scalar to ensure the estimates of the probabilities of belonging to a class or uniformly lower bounded by a positive constant. |
| epsout | If the probability of belonging of a data to a class is smaller than exp(epsout), this probbility is replaced by exp(epsout) for calculating the logLikelihood. If the probability is too weak for each class, the data is considered as an outlier. Defautl is -20. |

| | |
|---|---|
| alpha | A scalar between 1/2 and 1 used in the stepsequence for the Robbins-Monro method if methodMC='RobbinsMC'. |
| c | The constant in the stepsequence if methodMC='RobbinsMC' or methodMC='GradMC'. |
| w | The power for the weighted averaged Robbins-Monro algorithm if methodMC='RobbinsMC'. |
| epsilon | Stoping condition for the Weiszfeld algorithm. |
| criterion | The criterion for selecting the number of cluster. Can be 'ICL' (default) or 'BIC'. |
| methodMC | The method chosen to estimate robustly the variance. Can be 'RobbinsMC', 'GradMC' or 'FixMC'. |
| par | Is equal to T if the parallelization of the algorithm is allowed. |
| methodMCM | The method chosen for estimating the Median Covariation Matrix. Can be 'Gmedian' or 'Weiszfeld' |

## Value

A list with:

| | |
|---|---|
| bestresult | A list giving all the results fo the best clustering (chosen with respect to the selected criterion. |
| allresults | A list containing all the results. |
| ICL | The ICL criterion for all the number of classes selected. |
| BIC | The ICL criterion for all the number of classes selected. |
| data | The initial data. |
| nclust | A vector of positive integers giving the possible number of clusters. |
| Kopt | The number of clusters chosen by the selected criterion. |

For the lists bestresult and allresults[[k]]:

| | |
|---|---|
| centers | A matrix whose rows are the centers of the classes. |
| Sigma | A matrix containing all the variance of the classes |
| LogLike | The final LogLikelihood. |
| Pi | A matrix giving the probabilities of each data to belong to each class. |
| niter | The number of iterations of the EM algorithm. |
| initEM | A vector giving the initialized clustering if init='Mclust' or init='genie'. |
| prop | A vector giving the proportions of each classes. |
| outliers | A vector giving the detected outliers. |

## References

Cardot, H., Cenac, P. and Zitt, P-A. (2013). Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19, 18-43.

Cardot, H. and Godichon-Baggioni, A. (2017). Fast Estimation of the Median Covariation Matrix with Application to Online Robust Principal Components Analysis. *Test*, 26(3), 461-480

Vardi, Y. and Zhang, C.-H. (2000). The multivariate L1-median and associated data depth. *Proc. Natl. Acad. Sci. USA*, 97(4):1423-1426.

## See Also

See also Gen_MM, RMMplot and RobVar.

## Examples

```
## Not run:
ech <- Gen_MM(mu = matrix(c(rep(-2,3),rep(2,3),rep(0,3)),byrow = TRUE,nrow=3))
 X <- ech$X
 res <- RobMM(X , nclust=3)
 RMMplot(res,graph=c('Two_Dim'))

## End(Not run)
```

---

RobVar *RobVar*

---

## Description

Robust estimate of the variance

## Usage

```
RobVar(X, c=2, alpha=0.75, model='Gaussian', methodMCM='Weiszfeld',
                methodMC='Robbins' , mc_sample_size=1000, init=rep(0, ncol(X)),
                init_cov=diag(ncol(X)),
                epsilon=10^(-8), w=2, df=3, niterMC=50,
            cgrad=2, niterWeisz=50, epsWeisz=10^-8, alphaMedian=0.75, cmedian=2)
```

## Arguments

| | |
|---|---|
| X | A numeric matrix of whose rows correspond to observations. |
| c | A positive scalar giving the constant in the stepsequence of the Robbins-Monro or Gradient method if methodMC='RobbinsMC' or methodMC='GradMC'. Default is 2. |
| alpha | A scalar between 1/2 and 1 giving the power in the stepsequence for the Robbins-Monro algorithm is methodMC='RobbinsMC'. Default is 0.75. |
| model | A string character specifying the model: can be 'Gaussian' (default), 'Student' or 'Laplace'. |
| methodMCM | A string character specifying the method to estimate the Median Covariation Matrix. Can be 'Gmedian' or 'Weiszfeld' (defualt). |
| methodMC | A string character specifying the method to estimate robustly the variance. Can be 'Robbins' (default), 'Fix' or 'Grad'. |
| mc_sample_size | A positive integer giving the number of data simulated for the Monte-Carlo method. Default is 1000. |
| init | A numeric vector giving the initialization for estimating the median. |

| | |
|---|---|
| init_cov | A numeric matrix giving an initialization for estimating the Median Covariation Matrix. |
| epsilon | A positive scalar giving a stoping condition for algorithm. |
| w | A positive integer specifying the power for the weighted averaged Robbins-Monro algorithm if methodMC='RobbinsMC'. |
| df | An integer larger (or equal) than 3 specifying the degrees of freedom for the Student law if model='Student'. See also Gen_MM. Default is 3. |
| niterMC | An integer giving the number of iterations for iterative algorithms if the selected method is 'Grad' or 'Fix'. Default is 50. |
| cgrad | A numeric vector with positive values giving the stepsequence of the gradient algorithm for estimating the variance if methodMC='Grad'. Its length has to be equal to niter. |
| niterWeisz | A positive integer giving the maximum number of iterations for the Weiszfeld algorithms if methodMCM='Weiszfeld'. Default is 50. |
| epsWeisz | A stopping factor for the Weiszfeld algorithm. |
| alphaMedian | A scalar betwwen 1/2 and 1 giving the power of the stepsequence of the gradient algorithm for estimating the Median Covariation Matrix if methodMCM='Gmedian'. Default is 0.75. |
| cmedian | A positive scalar giving the constant in the stepsequence of the gradient algorithm for estimating the Median Covariation Matrix if methodMCM='Gmedian'. Default is 2. |

## Value

An object of class list with the following outputs:

| | |
|---|---|
| median | The median of X. |
| variance | The robust variance of X. |
| median | The Median Covariation Matrix of X. |

## References

Cardot, H., Cenac, P. and Zitt, P-A. (2013). Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19, 18-43.

Cardot, H. and Godichon-Baggioni, A. (2017). Fast Estimation of the Median Covariation Matrix with Application to Online Robust Principal Components Analysis. *Test*, 26(3), 461-480

Vardi, Y. and Zhang, C.-H. (2000). The multivariate L1-median and associated data depth. *Proc. Natl. Acad. Sci. USA*, 97(4):1423-1426.

## See Also

See also RobMM and Gen_MM.

## Examples

```
n <- 2000
d <- 5
Sigma <-diag(1:d)
mean <- rep(0,d)
X <- mvtnorm::rmvnorm(n,mean,Sigma)
RVar=RobVar(X)
```

# Index