Package 'RobRegression'

April 23, 2024

Type Package

Title Robust Multivariate Regression

Version 0.1.0

Description Robust methods for estimating the parameters of multivariate Gaussian linear models.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp, foreach, doParallel, mvtnorm,parallel,RSpectra ,
capushe, KneeArrower, fastmatrix, DescTools

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

RoxygenNote 7.1.2

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

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

Repository CRAN

Date/Publication 2024-04-23 09:00:02 UTC

R topics documented:

RobRegression-package 2
Robust_Mahalanobis_regression 3
Robust_regression 6
Robust_Variance 8

Index **10**

RobRegression-package *Robust Multivariate Regression*

Description

This Package focuses on multivariate robust Guassian linear regression. We provide a function Robust_Mahalanobis_regression which enables to obtain robust estimates of the parameters of Multivariate Gaussian Linear Models with the help of the Mahalanobis distance, using a Stochastic Gradient algorithm or a Fix point. This is based on the function Robust_Variance which allows to obtain robust estimation of the variance, and so, also for low rank matrices (see Godichon-Baggioni and RObin (2024) <doi:10.1007/s11222-023-10362-9>) Robust methods for estimating the parameters of multivariate Gaussian linear models. .

Details

Package:	RobRegression
Type:	Package
Title:	Robust Multivariate Regression
Version:	0.1.0
Authors@R:	c(person("Antoine","Godichon-Baggioni", role = c("aut", "cre","cph"), email = "antoine.godichon_bagg
Description:	Robust methods for estimating the parameters of multivariate Gaussian linear models.
License:	GPL (>= 2)
Encoding:	UTF-8
LazyData:	true
Imports:	Rcpp, foreach, doParallel, mvtnorm,parallel,RSpectra , capushe, KneeArrower, fastmatrix, DescTools
LinkingTo:	Rcpp, RcppArmadillo
NeedsCompilation:	yes
Roxygen:	list(markdown = True)
RoxygenNote:	7.1.2
Author:	Antoine Godichon-Baggioni [aut, cre, cph], Stéphane Robin [aut], Laure Sansonnet [aut]
Maintainer:	Antoine Godichon-Baggioni <antoine.godichon_baggioni@upmc.fr>
Archs:	x64

Index of help topics:

RobRegression-package Robust Multivariate Regression
Robust_Mahalanobis_regression
Robust_Mahalanobis_regression
Robust_Variance Robust_Variance
Robust_regression Robust_regression

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

Godichon-Baggioni, A. and Robin, S. (2024). Recursive ridge regression using second-order stochastic *algorithms. Computational Statistics & Data Analysis, 190, 107854.*

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

`Robust_Mahalanobis_regression`

Robust_Mahalanobis_regression

Description

We propose here a function which enables to provide a robust estimation of the parameters of Multivariate Gaussian Linear Models of the form $Y = X\beta + \epsilon$ where ϵ is a 0-mean Gaussian vector of variance Σ. In addition, one can aslo consider a low-rank variance of the form $\Sigma = C + \sigma I$ where σ is a positive scalar and C is a matrix of rank d. More precisely, the aim is to minimize the functional

$G_\lambda(\hat{\beta}) = \mathbb{E}\left(\|Y - X\hat{\beta}\|_{\Sigma^{-1}}\right) + \lambda\|\hat{\beta}\|^{\text{Ridge}}$.

Usage

```
Robust_Mahalanobis_regression(X, Y, alphaRM=0.66, alphareg=0.66, w=2, lambda=0,
                              creg='default', K=2:30, par=TRUE, epsilon=10^(-8),
                               method_regression='Offline', niter_regression=50,
                               cRM='default', mc_sample_size='default',
                               method_MCM='Weiszfeld', methodMC='Robbins',
                               niterMC=50, ridge=1, eps_vp=10^(-4), nlambda=50,
                               scale='none', tol=10^(-3))
```

Arguments

`X`	A (n, p)-matrix whose rows are the explaining data.
`Y`	A (n, q)-matrix whose rows are the variables to be explained.
`method_regression`	The method used for estimating the parameter. Should be `method_regression='Offline'` if the fix point algorithm is used, and `method_regression='Online'` if the (weighted) averaged stochastic gradient algorithm is used. Default is `'Offline'`.
`niter_regression`	The maximum number of regression iterations if the fix point algorithm is used, i.e. if `method_regression='Offline'`.

epsilon	Stoping condition for the fix point algorithm if method_regression='Offline'.
scale	If a scaling is used. scale='robust' should be used if a robust scaling of Y is desired. Default is 'none'.
ridge	The power of the penalty: i.e. should be 2 if the squared norm is considered or 1 if the norm is considered.
lambda	A vector giving the different studied penalizations. If lambda='default', would be a vector of preselected penalizations.
par	Is equal to T if the parallelization of the algorithm for estimating robustly the variance of the noise is allowed.
nlambda	The number of tested penalizations if lambda='default'.
alphaRM	A scalar between 1/2 and 1 used in the stepsequence if the Robbins-Monro algorithm is used, i.e. if methodMC='Robbins'. Default is 0.66.
alphareg	A scalar between 1/2 and 1 used in the stepsequence for stochastic gradient algorithm if method_regression='Online'. Default is 0.66.
w	The power for the weighted averaged algorithms if method_regression='Online' or if methodMC='Robbins'.
creg	The constant in the stepsequence if the averaged stochastic gradient algorithm is used, i.e. if method='Online'.
K	A vector containing the possible values of d. The good d is chosen with the help of a penatly criterion if the length of K is larger than 10. Default is ncol(X).
mc_sample_size	The number of data generated for the Monte-Carlo method for estimating robustly the eigenvalues of the variance.
method_MCM	The method chosen to estimate Median Covariation Matrix. Can be 'Weiszfeld' if the Weiszfeld algorithm is used, or 'ASGD' if one chooses the Averaged Stochastic Gradient Descent algorithm.
methodMC	The method chosen to estimate robustly the variance. Can be 'Robbins', 'Grad' or 'Fix'.
niterMC	The number of iterations for estimating robustly the variance of each class if methodMC='Fix' or methodMC='Grad'.
eps_vp	The minimum values for the estimates of the eigenvalues of the Variance can take. Default is 10^-4.
cRM	The constant in the stepsequence if the Robbins-Monro algorithm is used to robustly estimate the variance, i.e. if methodMC='Robbins'.
tol	A scalar that avoid numerical problems if method='Offline'. Default is 10^(-3).

Value

A list with:

beta	A (p, q)-matrix giving the estimation of the parameters of the MultivariateGaussian Linear Regression.
Residual_Variance	A (q, q)-matrix giving the estimation of the variance of the residuals.

`criterion` A vector giving the loss for the different chosen `lambda`. If `scale='robust'`, it is calculated on the scaled data.

`all_beta` A list containing the different estimation of the parameters (with respect to the different choices of `lambda`).

`lambda_opt` A scalar giving the selected `lambda`.

`variance_results`
A list giving the results on the variance of the noise obtained with the help of the function `Robust_Variance`. If `scale='robust'`, it is calculated on the scaled data. The details are given above.

Details of the list `variance_results`:

`Sigma` The robust estimation of the variance.

`invSigma` The robuste estimation of the inverse of the variance.

`MCM` The Median Covariation Matrix.

`eigenvalues` A vector containing the estimation of the $d+1$ main eigenvalues of the variance, where $d+1$ is the optimal choice belonging to `K`.

`MCM_eigenvalues`
A vector containing the estimation of the $d+1$ main eigenvalues of the Median Covariation Matrix, where $d+1$ is the optimal choice belonging to `K`.

`cap` The result given for capushe for selecting d if the length of `K` is larger than 10.

`reduction_results`
A list containing the results for all possible `K`.

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 `Robust_Variance`, `Robust_regression` and `RobRegression-package`.

Examples

```
p=5
q=10
n=2000
mu=rep(0,q)
Sigma=diag(c(q,rep(0.1,q-1)))
epsilon=mvtnorm::rmvnorm(n = n,mean = mu,sigma = Sigma)
X=mvtnorm::rmvnorm(n=n,mean=rep(0,p))
```

```
beta=matrix(rnorm(p*q),ncol=q)
Y=X %*% beta+epsilon
Res_reg=Robust_Mahalanobis_regression(X,Y,par=FALSE)
sum((Res_reg$beta-beta)^2)
```

Robust_regression *Robust_regression*

Description

This function gives robust estimates of the paramter of the Multivariate Linear regression with the help of the euclidean distance, or with the help of the Mahalanobis distance for some matrice Sigma. More precisely, the aim is to minimize

$G(\hat{\beta}) = \mathbb{E}[\|Y - X\hat{\beta}\|_\Sigma] + \lambda\|\hat{\beta}\|^{\text{ridge}}$.

Usage

```
Robust_regression(X,Y, Mat_Mahalanobis=diag(rep(1,ncol(Y))),
                  niter=50,lambda=0,c='default',method='Offline',
                  alpha=0.66,w=2,ridge=1,nlambda=50,
             init=matrix(runif(ncol(X)*ncol(Y))-0.5,nrow=ncol(X),ncol=ncol(Y)),
                  epsilon=10^(-8), Mahalanobis_distance = FALSE,
                  par=TRUE,scale='none',tol=10^(-3))
```

Arguments

X	A (n,p)-matrix whose raws are the explaining data.
Y	A (n,q)-matrix whose raws are the variables to be explained.
method	The method used for estimating the parameter. Should be `method='Offline'` if the fix point algorithm is used, and `'Online'` if the (weighted) averaged stochastic gradient algorithm is used. Default is `'Offline'`.
Mat_Mahalanobis	A (q,q)-matrix giving Σ for the Mahalanobis distance. Default is identity.
Mahalanobis_distance	A logical telling if the Mahalanobis distance is used. Default is `FALSE`.
scale	If a scaling is used. `scale='robust'` should be used if a robust scaling of `Y` is desired. Default is `'none'`
niter	The maximum number of iteration if `method='Offline'`.
init	A (p,q)-matrix which gives the initialization of the algorithm.
ridge	The power of the penalty: i.e should be `2` if the squared norm is considered or `1` if the norm is considered.
lambda	A vector giving the different studied penalizations. If `lambda='default'`, would be a vector of preselected penalizations.

nlambda	The number of tested penalizations if lambda='default'.
par	Is equal to TRUE if the parallelization of the algorithm for estimating robustly the variance of the noise is allowed.
c	The constant in the stepsequence if the averaged stochastic gradient algorithm, i.e if method='Online'.
alpha	A scalar between 1/2 and 1 used in the stepsequence for stochastic gradient algorithm if method='Online'.
w	The power for the weighted averaged Robbins-Monro algorithm if method='Online'.
epsilon	Stoping condition for the fix point algorithm if method='Offline'.
tol	A scalar that avoid numerical problems if method='Offline'. Default is 10^(-3).

Value

A list with:

beta	A (p,q)-matrix giving the estimation of the parameters.
criterion	A vector giving the loss for the different chosen lambda. If sale='robust', it is calculated on the scaled data.
all_beta	A list containing the different estimation of the parameters (with respect to the different coices of lambda).
lambda_opt	A scalar giving the selected lambda.

References

Godichon-Baggioni, A., Robin, S. and Sansonnet, L. (2023): A robust multivariate linear regression based on the Mahalanobis distance

See Also

See also Robust_Variance, Robust_Mahalanobis_regression and RobRegression-package.

Examples

```
p=5
q=10
n=2000
mu=rep(0,q)
epsilon=mvtnorm::rmvnorm(n = n,mean = mu)
X=mvtnorm::rmvnorm(n=n,mean=rep(0,p))
beta=matrix(rnorm(p*q),ncol=q)
Y=X %*% beta+epsilon
Res_reg=Robust_regression(X,Y)
sum((Res_reg$beta-beta)^2)
```

Robust_Variance *Robust_Variance*

Description

The aim is to provide a robust estimation of the variance for Guassian models with reduction dimension. More precisely we considering a q dimensional random vector whose variance can be written as $\Sigma = C + \sigma I$ where C is a matrix of rank d, with d possibly much smaller than q, $sigma$ is a positive scalar, and I is the identity matrix.

Usage

```
Robust_Variance(X,K=ncol(X),par=TRUE,alphaRM=0.75,
                c='default',w=2,mc_sample_size='default',
                methodMC='Robbins',niterMC=50,method_MCM='Weiszfeld',
                eps_vp=10^(-6))
```

Arguments

X	A matrix whose raws are the vector we want to estimate the variance.
K	A vector containing the possible values of d. The 'good' d is chosen with the help of a penatly criterion if the length of K is larger than 10. Default is `ncol(X)`.
par	Is equal to `TRUE` if the parallelization of the algorithm is allowed.
mc_sample_size	The number of data generated for the Monte-Carlo method for estimating robustly the eigenvalues of the variance.
methodMC	The method chosen to estimate robustly the variance. Can be `'Robbins'`, `'Grad'` or `'Fix'`. Default is `'Robbins'`.
niterMC	The number of iterations for estimating robustly the variance of each class if `methodMC='Fix'` or `methodMC='Grad'`.
method_MCM	The method chosen to estimate Median Covariation Matrix. Can be `'Weiszfeld'` or `'ASGD'`.
alphaRM	A scalar between 1/2 and 1 used in the stepsequence for the Robbins-Monro method if `methodMC='Robbins'`.
c	The constant in the stepsequence if `methodMC='Robbins'`.
w	The power for the weighted averaged Robbins-Monro algorithm if `methodMC='Robbins'`. Default is `2`.
eps_vp	The minimum values for the estimates of the eigenvalues of the Variance can take. Default is `10^-6`.

Value

A list with:

Sigma	The robust estimation of the variance.

`invSigma` The robuste estimation of the inverse of the variance.

`MCM` The Median Covariation Matrix.

`eigenvalues` A vector containing the estimation of the d+1 main eigenvalues of the variance, where d+1 is the optimal choice belong K.

`MCM_eigenvalues` A vector containing the estimation of the d+1 main eigenvalues of the Median Covariation Matrix, where d+1 is the optimal choice belong K.

`cap` The result given for capushe for selecting d if the length of K is larger than 10.

`reduction_results` A list containing the results for all possible K.

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 `Robust_Mahalanobis_regression`, `Robust_regression` and `RobRegression-package`.

Examples

```
q<-100
d<-10
n<-2000
Sigma<- diag(c(d:1,rep(0,q-d)))+ diag(rep(0.1,q))
X=mvtnorm::rmvnorm(n=n,sigma=Sigma)
RobVar = Robust_Variance(X,K=q)
sum((RobVar$Sigma-Sigma)^2)/q
```

Index

RobRegression-package, 2
Robust_Mahalanobis_regression, *2*, 3, *7*, *9*
Robust_regression, *5*, 6, *9*
Robust_Variance, *2*, *5*, *7*, 8