

Package: adaHuber (via r-universe)

October 10, 2024

Type Package

Title Adaptive Huber Estimation and Regression

Version 1.1

Date 2022-03-04

Description Huber-type estimation for mean, covariance and (regularized) regression. For all the methods, the robustification parameter tau is chosen by a tuning-free principle.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

URL <https://github.com/XiaoouPan/adaHuber>

SystemRequirements C++11

Imports Rcpp (>= 1.0.3)

LinkingTo Rcpp, RcppArmadillo (>= 0.9.850.1.0)

RoxygenNote 7.1.1

NeedsCompilation yes

Author Xiaoou Pan [aut, cre], Wen-Xin Zhou [aut]

Maintainer Xiaoou Pan <xip024@ucsd.edu>

Repository CRAN

Date/Publication 2022-03-09 07:50:02 UTC

Contents

adaHuber-package	2
adaHuber.cov	2
adaHuber.cv.lasso	3
adaHuber.lasso	5
adaHuber.mean	7
adaHuber.reg	8

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

adaHuber-package

adaHuber: Adaptive Huber Estimation and Regression

Description

Huber-type robust estimation for mean, covariance and (penalized) regression.

Author(s)

Xiaou Pan <xip024@ucsd.edu> and Wen-Xin Zhou <wez243@ucsd.edu>

References

Ke, Y., Minsker, S., Ren, Z., Sun, Q. and Zhou, W.-X. (2019). User-friendly covariance estimation for heavy-tailed distributions. *Statis. Sci.*, 34, 454-471.

Pan, X., Sun, Q. and Zhou, W.-X. (2021). Iteratively reweighted l1-penalized robust regression. *Electron. J. Stat.*, 15, 3287-3348.

Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. *J. Amer. Stat. Assoc.*, 115, 254-265.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. *Stat. Sinica*, 31, 2153-2177.

adaHuber.cov

Adaptive Huber Covariance Estimation

Description

Adaptive Huber covariance estimator from a data sample, with robustification parameter τ determined by a tuning-free principle.

Usage

```
adaHuber.cov(X, epsilon = 1e-04, iteMax = 500)
```

Arguments

X	An n by p data matrix.
epsilon	(optional) The tolerance level in the iterative estimation procedure. The problem is converted to mean estimation, and the stopping rule is the same as <code>adaHuber.mean</code> . The default value is $1e-4$.
iteMax	(optional) Maximum number of iterations. Default is 500.

Details

The observed data X is an n by p matrix. The distribution of each entry can be asymmetrix and/or heavy-tailed. The function outputs a robust estimator for the covariance matrix of X . For the input matrix X , both low-dimension ($p < n$) and high-dimension ($p > n$) are allowed.

Value

A list including the following terms will be returned:

means The Huber estimators for column means. A p -dimensional vector.

cov The Huber estimator for covariance matrix. A p by p matrix.

References

Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35, 73–101.

Ke, Y., Minsker, S., Ren, Z., Sun, Q. and Zhou, W.-X. (2019). User-friendly covariance estimation for heavy-tailed distributions. *Statis. Sci.*, 34, 454-471.

See Also

[adaHuber.mean](#) for adaptive Huber mean estimation.

Examples

```
n = 100
p = 5
X = matrix(rt(n * p, 3), n, p)
fit.cov = adaHuber.cov(X)
fit.cov$means
fit.cov$cov
```

adaHuber.cv.lasso

Cross-Validated Regularized Adaptive Huber Regression.

Description

Sparse regularized adaptive Huber regression with "lasso" penalty. The function implements a localized majorize-minimize algorithm with a gradient-based method. The regularization parameter λ is selected by cross-validation, and the robustification parameter τ is determined by a tuning-free principle.

Usage

```
adaHuber.cv.lasso(
  X,
  Y,
  lambdaSeq = NULL,
  kfolds = 5,
  numLambda = 50,
  phi0 = 0.01,
  gamma = 1.2,
  epsilon = 0.001,
  iteMax = 500
)
```

Arguments

X	A n by p design matrix. Each row is a vector of observation with p covariates.
Y	An n -dimensional response vector.
lambdaSeq	(optional) A sequence of candidate regularization parameters. If unspecified, a reasonable sequence will be generated.
kfolds	(optional) Number of folds for cross-validation. Default is 5.
numLambda	(optional) Number of λ values for cross-validation if lambdaSeq is unspecified. Default is 50.
phi0	(optional) The initial quadratic coefficient parameter in the local adaptive majorize-minimize algorithm. Default is 0.01.
gamma	(optional) The adaptive search parameter (greater than 1) in the local adaptive majorize-minimize algorithm. Default is 1.2.
epsilon	(optional) A tolerance level for the stopping rule. The iteration will stop when the maximum magnitude of the change of coefficient updates is less than epsilon. Default is 0.001.
iteMax	(optional) Maximum number of iterations. Default is 500.

Value

An object containing the following items will be returned:

`coef` A $(p + 1)$ vector of estimated sparse regression coefficients, including the intercept.

`lambdaSeq` The sequence of candidate regularization parameters.

`lambda` Regularization parameter selected by cross-validation.

`tau` The robustification parameter calibrated by the tuning-free principle.

`iteration` Number of iterations until convergence.

`phi` The quadratic coefficient parameter in the local adaptive majorize-minimize algorithm.

References

- Pan, X., Sun, Q. and Zhou, W.-X. (2021). Iteratively reweighted l1-penalized robust regression. *Electron. J. Stat.*, 15, 3287-3348.
- Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. *J. Amer. Statist. Assoc.*, 115, 254-265.
- Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. *Stat. Sinica*, 31, 2153-2177.

See Also

See [adaHuber.lasso](#) for regularized adaptive Huber regression with a specified *lambda*.

Examples

```
n = 100; p = 200; s = 5
beta = c(rep(1.5, s + 1), rep(0, p - s))
X = matrix(rnorm(n * p), n, p)
err = rt(n, 2)
Y = cbind(rep(1, n), X) %*% beta + err

fit.lasso = adaHuber.cv.lasso(X, Y)
beta.lasso = fit.lasso$coef
```

adaHuber.lasso

Regularized Adaptive Huber Regression

Description

Sparse regularized Huber regression models in high dimensions with ℓ_1 (lasso) penalty. The function implements a localized majorize-minimize algorithm with a gradient-based method.

Usage

```
adaHuber.lasso(
  X,
  Y,
  lambda = 0.5,
  tau = 0,
  phi0 = 0.01,
  gamma = 1.2,
  epsilon = 0.001,
  iteMax = 500
)
```

Arguments

<code>X</code>	A n by p design matrix. Each row is a vector of observation with p covariates.
<code>Y</code>	An n -dimensional response vector.
<code>lambda</code>	(optional) Regularization parameter. Must be positive. Default is 0.5.
<code>tau</code>	(optional) The robustness parameter. If not specified or the input value is non-positive, a tuning-free principle is applied. Default is 0 (hence, tuning-free).
<code>phi0</code>	(optional) The initial quadratic coefficient parameter in the local adaptive majorize-minimize algorithm. Default is 0.01.
<code>gamma</code>	(optional) The adaptive search parameter (greater than 1) in the local adaptive majorize-minimize algorithm. Default is 1.2.
<code>epsilon</code>	(optional) Tolerance level of the gradient-based algorithm. The iteration will stop when the maximum magnitude of all the elements of the gradient is less than <code>tol</code> . Default is 1e-03.
<code>iteMax</code>	(optional) Maximum number of iterations. Default is 500.

Value

An object containing the following items will be returned:

`coef` A $(p + 1)$ vector of estimated sparse regression coefficients, including the intercept.
`tau` The robustification parameter calibrated by the tuning-free principle (if the input is non-positive).
`iteration` Number of iterations until convergence.
`phi` The quadratic coefficient parameter in the local adaptive majorize-minimize algorithm.

References

Pan, X., Sun, Q. and Zhou, W.-X. (2021). Iteratively reweighted l1-penalized robust regression. *Electron. J. Stat.*, 15, 3287-3348.

Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. *J. Amer. Statist. Assoc.*, 115 254-265.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. *Stat. Sinica*, 31, 2153-2177.

See Also

See [adaHuber.cv.lasso](#) for regularized adaptive Huber regression with cross-validation.

Examples

```
n = 200; p = 500; s = 10
beta = c(rep(1.5, s + 1), rep(0, p - s))
X = matrix(rnorm(n * p), n, p)
err = rt(n, 2)
Y = cbind(rep(1, n), X) %*% beta + err

fit.lasso = adaHuber.lasso(X, Y, lambda = 0.5)
beta.lasso = fit.lasso$coef
```

adaHuber.mean

Adaptive Huber Mean Estimation

Description

Adaptive Huber mean estimator from a data sample, with robustification parameter τ determined by a tuning-free principle.

Usage

```
adaHuber.mean(X, epsilon = 1e-04, iteMax = 500)
```

Arguments

X	An n -dimensional data vector.
epsilon	(optional) The tolerance level in the iterative estimation procedure, iteration will stop when $ \mu_{new} - \mu_{old} < \epsilon$. The default value is 1e-4.
iteMax	(optional) Maximum number of iterations. Default is 500.

Value

A list including the following terms will be returned:

mu The Huber mean estimator.

tau The robustness parameter determined by the tuning-free principle.

iteration The number of iterations in the estimation procedure.

References

Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35, 73–101.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. *Stat. Sinica*, 31, 2153-2177.

Examples

```
n = 1000
mu = 2
X = rt(n, 2) + mu
fit.mean = adaHuber.mean(X)
fit.mean$mu
```

adaHuber.reg

*Adaptive Huber Regression***Description**

Adaptive Huber regression from a data sample, with robustification parameter τ determined by a tuning-free principle.

Usage

```
adaHuber.reg(
  X,
  Y,
  method = c("standard", "adaptive"),
  epsilon = 1e-04,
  iteMax = 500
)
```

Arguments

X	A n by p design matrix. Each row is a vector of observation with p covariates. Number of observations n must be greater than number of covariates p .
Y	An n -dimensional response vector.
method	(optional) A character string specifying the method to calibrate the robustification parameter τ . Two choices are "standard"(default) and "adaptive". See Wang et al.(2021) for details.
epsilon	(optional) Tolerance level of the gradient descent algorithm. The iteration will stop when the maximum magnitude of all the elements of the gradient is less than tol. Default is 1e-04.
iteMax	(optional) Maximum number of iterations. Default is 500.

Value

An object containing the following items will be returned:

coef A $(p + 1)$ -vector of estimated regression coefficients, including the intercept.
 tau The robustification parameter calibrated by the tuning-free principle.
 iteration Number of iterations until convergence.

References

Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35, 73–101.
 Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. *J. Amer. Statist. Assoc.*, 115, 254-265.
 Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. *Stat. Sinica*, 31, 2153-2177.

Examples

```
n = 200
p = 10
beta = rep(1.5, p + 1)
X = matrix(rnorm(n * p), n, p)
err = rt(n, 2)
Y = cbind(1, X) %*% beta + err

fit.huber = adaHuber.reg(X, Y, method = "standard")
beta.huber = fit.huber$coef

fit.adahuber = adaHuber.reg(X, Y, method = "adaptive")
beta.adahuber = fit.adahuber$coef
```

Index

adaHuber-package, 2
adaHuber.cov, 2
adaHuber.cv.lasso, 3, 6
adaHuber.lasso, 5, 5
adaHuber.mean, 3, 7
adaHuber.reg, 8