Package ‘HCTR’

November 22, 2019

Title Higher Criticism Tuned Regression
Version 0.1.1
Description A novel searching scheme for tuning parameter in high-dimensional penalized regression. We propose a new estimate of the regularization parameter based on an estimated lower bound of the proportion of false null hypotheses (Meinshausen and Rice (2006) <doi:10.1214/009053605000000741>). The bound is estimated by applying the empirical null distribution of the higher criticism statistic, a second-level significance testing, which is constructed by dependent p-values from a multi-split regression and aggregation method (Jeng, Zhang and Tzeng (2019) <doi:10.1080/01621459.2018.1518236>). An estimate of tuning parameter in penalized regression is decide corresponding to the lower bound of the proportion of false null hypotheses. Different penalized regression methods are provided in the multi-split algorithm.

Depends R (>= 3.4.0)
Imports glmnet (>= 2.0-18), harmonicmeanp (>= 3.0), MASS, ncvreg (>= 3.11-1), Rdpack (>= 0.11-0), stats
RdMacros Rdpack
License GPL-2
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no
Author Tao Jiang [aut, cre]
Maintainer Tao Jiang <tjiang8@ncsu.edu>
Repository CRAN
Date/Publication 2019-11-22 21:50:09 UTC

R topics documented:

  bounding.seq ................................................................. 2
  est.lambda ................................................................. 3
Description

Calculates bounding sequence of higher criticism for proportion estimator using p-values

Usage

bounding.seq(p.value, alpha)

Arguments

p.value A matrix of p-values from permutation: row is from each permutation; column is from each variable.

alpha Probability of Type I error for bounding sequence, the default value is $1 / \sqrt{\log(p)}$, where $p$ is number of p-values in each permutation.

Value

A bounding value of higher criticism with (1 - alpha) confidence.

References


Examples

```r
set.seed(10)
X <- matrix(runif(n = 10000, min = 0, max = 1), nrow = 100)
result <- bounding.seq(p.value = X)
```
**est.lambda**

*Estimated Lambda*

---

**Description**

Estimate upper and lower bound of new tuning region of regularization parameter Lambda.

**Usage**

```
est.lambda(cv.fit, pihat, p, cov.num = 0)
```

**Arguments**

- `cv.fit`: An object of either class "cv.glmnet" from glmnet::cv.glmnet() or class "cv.ncvreg" from ncvreg::cv.ncvreg(), which is a list generated by a cross-validation fit.
- `pihat`: Estimated proportion from HCTR::est.prop().
- `p`: Total number of variables, except for covariates.
- `cov.num`: Number of covariates in model, default is 0. Covariate matrix, W, is assumed on the left side of variable matrix, X. The column index of covariates are before those of variables.

**Value**

A list of (1) lambda.max, upper bound of new tuning region; (2) lambda.min, lower bound of new tuning region.

**Examples**

```
set.seed(10)
X <- matrix(rnorm(20000), nrow = 100)
beta <- rep(0, 200)
beta[1:100] <- 5
Y <- MASS::mvrnorm(n = 1, mu = X%*%beta, Sigma = diag(100))
fit <- glmnet::cv.glmnet(x = X, y = Y)
pihat <- 0.01
result <- est.lambda(cv.fit = fit, pihat = pihat, p = ncol(X))
```

---

**est.prop**

*Proportion Estimation*

---

**Description**

Estimates false null hypothesis Proportion from multiple p-values using higher criticism test estimator.
Usage

\texttt{est.prop(p.value, cn, adj = TRUE)}

Arguments

- **p.value**: A sequence of p-values from test data, not including p-values from covariates.
- **cn**: A value of bounding sequence generated by HCTR::bounding.seq().
- **adj**: A boolean algebra to decide whether to use adjusted Higher Criticism test statistic, the default value is TRUE.

Value

An estimated proportion of false null hypothesis.

References


Examples

```r
set.seed(10)
X <- matrix(runif(n = 10000, min = 0, max = 1), nrow = 100)
result <- bounding.seq(p.value = X)
Y <- matrix(runif(n = 100, min = 0, max = 1), nrow = 100)
test <- est.prop(p.value = Y, cn = result)
```

---

**final.selection**

*Final Selection*

Description

Returns the index of final selected variables in the final chosen model.

Usage

\texttt{final.selection(cv.fit, pihat, p, cov.num = 0)}

Arguments

- **cv.fit**: An object of either class "cv.glmnet" from glmnet::cv.glmnet() or class "cv.ncvreg" from ncvreg::cv.ncvreg(), which is a list generated by a cross-validation fit.
- **pihat**: Estimated proportion from HCTR::est.prop().
- **p**: Total number of variables, except for covariates.
- **cov.num**: Number of covariates in model, default is 0. Covariate matrix, \( W \), is assumed on the left side of variable matrix, \( X \). The column index of covariates are before those of variables.
A sequence of index of final selected variables in the final chosen model.

Examples

```r
set.seed(10)
X <- matrix(rnorm(20000), nrow = 100)
beta <- rep(0, 200)
beta[1:100] <- 5
Y <- MASS::mvrnorm(n = 1, mu = X%*%beta, Sigma = diag(100))
fit <- glmnet::cv.glmnet(x = X, y = Y)
pihat <- 0.01
result <- est.lambda(cv.fit = fit, pihat = pihat, p = ncol(X))
lambda.seq <- seq(from = result$lambda.min, to = result$lambda.max, length.out = 100)
# Note: The lambda sequences in glmnet and ncvreg are different.
fit2 <- glmnet::cv.glmnet(x = X, y = Y, lambda = lambda.seq)
result2 <- final.selection(cv.fit = fit2, pihat = 0.01, p = ncol(X))
```

**Description**

Calculates p-values in high-dimensional linear models using multi-split method

**Usage**

`highdim.p(Y, X, W = NULL, type, B = 100, fold.num)`

**Arguments**

- **Y**
  A numeric response vector, containing nob variables.

- **X**
  An input matrix, of dimension nob x nvars.

- **W**
  A covariate matrix, of dimension nob x ncors, default is NULL.

- **type**
  Penalized regression type, valid parameters include "Lasso", "AdaLasso", "SCAD", and "MCP".

- **B**
  Multi-split times, default is 100.

- **fold.num**
  The number of cross validation folds.

**Value**

A list of objects containing: (1) harmonic mean p-values; (2) original p-values; (3) index of selected samples; (4) index of selected variables.
Examples

```r
set.seed(10)
X <- matrix(rnorm(20000), nrow = 100)
beta <- rep(0, 200)
beta[1:100] <- 5
Y <- MASS::mvrnorm(n = 1, mu = X %*% beta, Sigma = diag(100))
result <- highdim.p(Y = Y, X = X, type = "Lasso", B = 2, fold.num = 10)
```

---

**multi.adlasso**  
*Multi-split Adaptive Lasso*

**Description**

Multi-splitted variable selection using Adaptive Lasso

**Usage**

```r
multi.adlasso(X, Y, covar.num = NULL, fold.num)
```

**Arguments**

- **X**: An input matrix, of dimension nobs x nvars.
- **Y**: A numeric response vector, containing nobs variables.
- **covar.num**: Number of covariates in model, default is NULL.
- **fold.num**: The number of cross validation folds.

**Value**

A list of two numeric objects of index of (1) selected and (2) unselected variables.

---

**multi.lasso**  
*Multi-split Lasso*

**Description**

Multi-splitted variable selection using Lasso

**Usage**

```r
multi.lasso(X, Y, p.fac = NULL, fold.num)
```
multi.mcp

Arguments

X An input matrix, of dimension nobs x nvars.
Y A numeric response vector, containing nobs variables.
p.fac A sequence of penalty factor applied on each variable.
fold.num The number of cross validation folds.

Value

A list of two numeric objects of index of (1) selected and (2) unselected variables.

multi.mcp Multi-split MCP

Description

Multi-splitted variable selection using MCP

Usage

multi.mcp(X, Y, p.fac = NULL, fold.num)

Arguments

X An input matrix, of dimension nobs x nvars.
Y A numeric response vector, containing nobs variables.
p.fac A sequence of penalty factor applied on each variable.
fold.num The number of cross validation folds.

Value

A list of two numeric objects of index of (1) selected and (2) unselected variables.
multi.scad \hspace{1cm} \textit{Multi-split SCAD}

Description
Multi-splitted variable selection using SCAD

Usage
\begin{verbatim}
multi.scad(X, Y, p.fac = NULL, fold.num)
\end{verbatim}

Arguments
\begin{verbatim}
X \hspace{1cm} \text{An input matrix, of dimension noobs x nvars.}
Y \hspace{1cm} \text{A numeric response vector, containing noobs variables.}
p.fac \hspace{1cm} \text{A sequence of penalty factor applied on each variable.}
fold.num \hspace{1cm} \text{The number of cross validation folds.}
\end{verbatim}

Value
A list of two numeric objects of index of (1) selected and (2) unselected variables.

pmpv \hspace{1cm} \textit{Permutation p-values}

Description
Calculates

Usage
\begin{verbatim}
pmpv(Y, X, W = NULL, type, B = 100, fold.num = 10, perm.num = 1000)
\end{verbatim}

Arguments
\begin{verbatim}
Y \hspace{1cm} \text{A numeric response vector, containing noobs variables.}
X \hspace{1cm} \text{An input matrix, of dimension noobs x nvars.}
W \hspace{1cm} \text{A covariate matrix, of dimension noobs x ncors, default is NULL.}
type \hspace{1cm} \text{Penalized regression type, valid parameters include "Lasso", "AdaLasso", "SCAD", and "MCP".}
B \hspace{1cm} \text{Multi-split times, default is 100.}
fold.num \hspace{1cm} \text{The number of cross validation folds, default is 10.}
perm.num \hspace{1cm} \text{Permutation times, default is 1000.}
\end{verbatim}
**Value**

A matrix containing harmonic mean p-values from permutation.

**Examples**

```r
set.seed(10)
X <- matrix(rnorm(20000), nrow = 100)
beta <- rep(0, 200)
beta[1:100] <- 5
Y <- MASS::mvrnorm(n = 1, mu = X%*%beta, Sigma = diag(100))
result <- pmpv(Y=Y, X=X, type = "Lasso", B = 2, fold.num = 10, perm.num = 10)
```
Index

bounding.seq, 2
est.lambda, 3
est.prop, 3
final.selection, 4
highdim.p, 5
multi.adlasso, 6
multi.lasso, 6
multi.mcp, 7
multi.scad, 8
pmpv, 8