Package ‘SIHR’

August 10, 2022

Type Package

Title Statistical Inference in High Dimensional Regression

Version 1.0.1

Author Prabrisha Rakshit, Zhenyu Wang, Zijian Guo, Tony Cai

Maintainer Zijian Guo <zijguo@stat.rutgers.edu>

Description Inference procedures in the high-dimensional setting for
(1) linear functionals in generalized linear regression ('Cai et al.' (2019) <arXiv:1904.12891>, 'Guo et al.' (2020) <arXiv:2012.07133>, 'Cai et al.' (2021)),
(2) individual treatment effects in generalized linear regression,
(3) quadratic functionals in linear regression ('Guo et al.' (2019) <arXiv:1909.01503>).

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.0

URL https://github.com/prabrishar1/SIHR

Imports CVXR, glmnet, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-10 07:30:13 UTC

R topics documented:

ITE *Inference for difference of linear combinations of the regression vectors in high dimensional generalized linear regressions*

Description

Computes the bias-corrected estimator of the difference of linearcombinations of the regression vectors for the high dimensional generalized linear regressions and the corresponding standard error.

Usage

```
ITE(
  X1,
  y1,
  X2,
  y2,
  loading.mat,
  model = "linear",
  intercept = TRUE,
  intercept.loading = FALSE,
  beta.init1 = NULL,
  beta.init2 = NULL,
  lambda = NULL,
  mu = NULL,
  init.step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`X1`	Design matrix for the first sample, of dimension n_1 x p
`y1`	Outcome vector for the first sample, of length n_1
`X2`	Design matrix for the second sample, of dimension n_2 x p
`y2`	Outcome vector for the second sample, of length n_1
`loading.mat`	Loading matrix, nrow=p, each column corresponds to a loading of interest
`model`	The high dimensional regression model, either `linear` or `logistic` or `logistic alternative` or `probit`
`intercept`	Should intercept(s) be fitted for the initial estimators (default = `TRUE`)
`intercept.loading`	Should intercept be included for the `loading` (default = `FALSE`)
`beta.init1`	The initial estimator of the regression vector for the 1st data (default = `NULL`)

`beta.init2`	The initial estimator of the regression vector for the 2nd data (default = `NULL`)
`lambda`	lambda The tuning parameter in fitting model (default = `NULL`)
`mu`	The dual tuning parameter used in the construction of the projection direction (default = `NULL`)
`init.step`	The initial step size used to compute `mu`; if set to `NULL` it is computed to be the number of steps (`maxiter`) to obtain the smallest `mu`
`resol`	The factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	Maximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)
`alpha`	Level of significance to construct two-sided confidence interval (default = 0.05)
`verbose`	Should intermediate message(s) be printed (default = `TRUE`)

Value

`est.plugin.vec`	The vector of plugin(biased) estimators for the linear combination of regression coefficients, length of `ncol(loading.mat)`; corresponding to different column in `loading.mat`
`est.debias.vec`	The vector of bias-corrected estimators for the linear combination of regression coefficients, length of `ncol(loading.mat)`; corresponding to different column in `loading.mat`
`se.vec`	The vector of standard errors of the bias-corrected estimators, length of `ncol(loading.mat)`; corresponding to different column in `loading.mat`
`ci.mat`	The matrix of two.sided confidence interval for the linear combination, of dimension `ncol(loading.mat) x 2`; the row corresponding to different column in `loading.mat`
`prob.debias.vec`	The vector of bias-corrected estimators after probability transformation, length of `ncol(loading.mat)`; corresponding to different column in loading.mat. The value would be `NULL` for non-logistic model.
`prob.se.vec`	The vector of standard errors of the bias-corrected estimators after probability transformation, length of `ncol(loading.mat)`; corresponding to different column in `loading.mat`. The value would be `NULL` for non-logistic model.

Examples

```
X1 = matrix(rnorm(100*10), nrow=100, ncol=10)
y1 = -0.5 + X1[,1] * 0.5 + X1[,2] * 1 + rnorm(100)
X2 = matrix(rnorm(90*10), nrow=90, ncol=10)
y2 = -0.4 + X2[,1] * 0.48 + X2[,2] * 1.1 + rnorm(90)
loading1 = c(1, 1, rep(0,8))
loading2 = c(-0.5, -1, rep(0,8))
loading.mat = cbind(loading1, loading2)
Est = ITE(X1, y1, X2, y2, loading.mat, model="linear")
```

```
## compute confidence intervals
ci(Est, alpha=0.05, alternative="two.sided")

## summary statistics
summary(Est)
```

LF *Inference for linear combination of the regression vector in high dimensional generalized linear regression*

Description

Inference for linear combination of the regression vector in high dimensional generalized linear regression

Usage

```
LF(
  X,
  y,
  loading.mat,
  model = c("linear", "logistic", "logistic_alternative", "probit"),
  intercept = TRUE,
  intercept.loading = FALSE,
  beta.init = NULL,
  lambda = NULL,
  mu = NULL,
  init.step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

X	Design matrix, of dimension n x p
y	Outcome vector, of length n
loading.mat	Loading matrix, nrow=p, each column corresponds to a loading of interest
model	The high dimensional regression model, either `linear` or `logistic` or `logistic_alternative` or `probit`
intercept	Should intercept be fitted for the initial estimator (default = `TRUE`)
intercept.loading	Should intercept be included for the loading (default = `FALSE`)
beta.init	The initial estimator of the regression vector (default = `NULL`)

`lambda`	The tuning parameter in fitting model (default = `NULL`)
`mu`	The dual tuning parameter used in the construction of the projection direction (default = `NULL`)
`init.step`	The initial step size used to compute `mu`; if set to `NULL` it is computed to be the number of steps (`maxiter`) to obtain the smallest `mu`
`resol`	The factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	Maximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)
`alpha`	Level of significance to construct two-sided confidence interval (default = 0.05)
`verbose`	Should intermediate message(s) be printed (default = `TRUE`)

Value

`est.plugin.vec`	The vector of plugin(biased) estimators for the linear combination of regression coefficients, length of `ncol(loading.mat)`; each corresponding to a loading of interest
`est.debias.vec`	The vector of bias-corrected estimators for the linear combination of regression coefficients, length of `ncol(loading.mat)`; each corresponding to a loading of interest
`se.vec`	The vector of standard errors of the bias-corrected estimators, length of `ncol(loading.mat)`; each corresponding to a loading of interest
`ci.mat`	The matrix of two.sided confidence interval for the linear combination, of dimension `ncol(loading.mat)` x 2; each row corresponding to a loading of interest
`proj.mat`	The matrix of projection directions; each column corresponding to a loading of interest

Examples

```
X = matrix(rnorm(100*10), nrow=100, ncol=10)
y = -0.5 + X[,1] * 0.5 + X[,2] * 1 + rnorm(100)
loading1 = c(1, 1, rep(0, 8))
loading2 = c(-0.5, -1, rep(0, 8))
loading.mat = cbind(loading1, loading2)
Est = LF(X, y, loading.mat, model="linear")

## compute confidence intervals
ci(Est, alpha=0.05, alternative="two.sided")

## summary statistics
summary(Est)
```

QF — *Inference for quadratic forms of the regression vector in high dimensional linear and logistic regressions*

Description

Inference for quadratic forms of the regression vector in high dimensional linear and logistic regressions

Usage

```
QF(
  X,
  y,
  G,
  A = NULL,
  model = c("linear", "logistic", "logistic_alternative", "probit"),
  intercept = TRUE,
  tau.vec = c(0, 0.5, 1),
  beta.init = NULL,
  lambda = NULL,
  mu = NULL,
  init.step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`X`	Design matrix, of dimension n x p
`y`	Outcome vector, of length n
`G`	The set of indices, `G` in the quadratic form
`A`	The matrix A in the quadratic form, of dimension $\|G\|\times\|G\|$. If `NULL` A would be set as the $\|G\|\times\|G\|$ submatrix of the population covariance matrix corresponding to the index set `G` (default = `NULL`)
`model`	The high dimensional regression model, either `linear` or `logistic` or `logistic_alternative` or `probit`
`intercept`	Should intercept be fitted for the initial estimator (default = `TRUE`)
`tau.vec`	The vector of enlargement factors for asymptotic variance of the bias-corrected estimator to handle super-efficiency (default = $c(0, 0.5, 1)$)
`beta.init`	The initial estimator of the regression vector (default = `NULL`)
`lambda`	The tuning parameter in fitting model (default = `NULL`)

`mu`	The dual tuning parameter used in the construction of the projection direction (default = `NULL`)
`init.step`	The initial step size used to compute `mu`; if set to `NULL` it is computed to be the number of steps (< `maxiter`) to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = `NULL`)
`resol`	Resolution or the factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	aximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)
`alpha`	Level of significance to construct two-sided confidence interval (default = 0.05)
`verbose`	Should intermediate message(s) be printed (default = `TRUE`)

Value

`est.plugin`	The plugin(biased) estimator for the quadratic form of the regression vector restricted to `G`
`est.debias`	The bias-corrected estimator of the quadratic form of the regression vector
`se.vec`	The vector of standard errors of the bias-corrected estimator, length of `tau.vec`; corrsponding to different values of `tau.vec`
`ci.mat`	The matrix of two.sided confidence interval for the quadratic form of the regression vector; row corresponds to different values of `tau.vec`
`proj`	The projection direction

Examples

```
X = matrix(rnorm(100*10), nrow=100, ncol=10)
y = X[,1] * 0.5 + X[,2] * 1 + rnorm(100)
G = c(1,2)
A = matrix(c(1.5, 0.8, 0.8, 1.5), nrow=2, ncol=2)
Est = QF(X, y, G, A, model="linear")
## compute confidence intervals
ci(Est, alpha=0.05, alternative="two.sided")

## summary statistics
summary(Est)
```

Index