Package 'SIHR'

June 30, 2022

Type Package

Title Statistical Inference in High Dimensional Regression

Version 1.0.0

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) quadratic functionals in linear regression ('Guo et al.' (2019) <arXiv:1909.01503>)
(3) individual treatment effects in generalized linear regression.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

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

Imports CVXR,
glmnet,
stats

R topics documented:

confint.ITE *Confidence Intervals for Bias-corrected ITE Estimators*

Description

Computes confidence intervals for bias-corrected estimators; Each row corresponds to a loading.

Usage

```
## S3 method for class 'ITE'
confint(obj, alpha = 0.05, alternative = c("two.sided", "less", "greater"))
```

Arguments

obj	An object of class 'ITE', a result of a call to 'ITE'
alpha	Level of significance to construct confidence interval (default=0.05)
alternative	Indicates the alternative hypothesis to construct confidence interval and must be one of "two.sided" (default), "less", or "greater".

Value

A matrix with columns giving lower and upper confidence limits for bias-corrected estimators.

Examples

```
## Not run:
##-- Continuing the ITE(.) example:
ci = confint(Est)
ci

## End(Not run)
```

confint.LF *Confidence Intervals for Bias-corrected LF Estimators*

Description

Computes confidence intervals for bias-corrected estimators; Each row corresponds to a loading.

Usage

```
## S3 method for class 'LF'
confint(obj, alpha = 0.05, alternative = c("two.sided", "less", "greater"))
```

Arguments

obj	An object of class 'LF', a result of a call to 'LF'
alpha	Level of significance to construct confidence interval (default=0.05)
alternative	Indicates the alternative hypothesis to construct confidence interval and must be one of "two.sided" (default), "less", or "greater".

Value

A matrix with columns giving lower and upper confidence limits for bias-corrected estimators.

Examples

```
## Not run:
##-- Continuing the LF(.) example:
ci = confint(Est)
ci

## End(Not run)
```

`confint.QF` *Confidence Intervals for Bias-corrected QF Estimators*

Description

Computes confidence intervals for bias-corrected estimators; Each row corresponds to a tau value.

Usage

```
## S3 method for class 'QF'
confint(obj, alpha = 0.05, alternative = c("two.sided", "less", "greater"))
```

Arguments

`obj`	An object of class ‘QF‘, a result of a call to ‘QF‘
`alpha`	Level of significance to construct confidence interval (default=0.05)
`alternative`	Indicates the alternative hypothesis to construct confidence interval and must be one of "two.sided" (default), "less", or "greater".

Value

A matrix with columns giving lower and upper confidence limits for bias-corrected estimators, with rows corresponding to different tau.

Examples

```
## Not run:
##-- Continuing the QF(.) example:
ci = confint(Est)
ci

## End(Not run)
```

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 = TRUE,
  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 = TRUE)
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

Examples

```
X1 = matrix(rnorm(100*120), nrow=100, ncol=120)
y1 = -0.5 + X1[,1] * 0.5 + X1[,2] * 1 + rnorm(100)
X2 = matrix(rnorm(90*120), nrow=90, ncol=120)
y2 = -0.4 + X2[,1] * 0.48 + X2[,2] * 1.1 + rnorm(90)
loading.mat = cbind(c(1, 1, rep(0, 118), c(-0.5, -1, rep(0, 118))))
Est = ITE(X1, y1, X2, y2, loading.mat, model="linear")
Est$est.plugin.vec ## plugin(biased) estimators
Est$est.debias.vec ## bias-corrected estimators
Est$se.vec ## standard errors for bias-corrected estimators
Est$ci.mat ## two-sided confidence interval for bias-corrected estimators
## Not run:
summary(Est)

## End(Not run)
```

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 = TRUE,
  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 = TRUE)
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*120), nrow=100, ncol=120)
y = -0.5 + X[,1] * 0.5 + X[,2] * 1 + rnorm(100)
loading.mat = cbind(c(1, 1, rep(0, 118), c(-0.5, -1, rep(0, 118))))
Est = LF(X, y, loading.mat, model="linear")
Est$est.plugin.vec ## plugin(biased) estimators
Est$est.debias.vec ## bias-corrected estimators
Est$se.vec ## standard errors for bias-corrected estimators
Est$ci.mat ## two-sided confidence interval for bias-corrected estimators
## Not run:
summary(Est)

## End(Not run)
```

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"),
  intercept = TRUE,
  tau.vec = c(0.5, 1),
  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
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 = $c(0.5, 1)$)
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*120), nrow=100, ncol=120)
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")
Est$est.plugin ## plugin(biased) estimator
Est$est.debias ## bias-corrected estimator
Est$se.vec ## standard errors for bias-corrected estimators for each tau
```

```
Est$ci.mat ## two-sided confidence interval for bias-corrected estimator for each tau
## Not run:
summary(Est)

## End(Not run)
```

---

summary.ITE *Summarizing ITE*

Description

'summary' method for class 'ITE'

Usage

```
## S3 method for class 'ITE'
summary(obj)
```

Arguments

obj	An object of class 'ITE', a result of a call to 'ITE'

Value

The function 'summary.ITE' computes and returns a list of summary statistics of ITE given 'obj'

output.est	a $ncol(loading.mat)$ x 7 matrix with columns for the loading, plugin(biased) estimators, bias-corrected estimators, its standard error, z-statistic, corresponding (two-sided) p-value and significance stars; Each row corresponds to each loading.

Examples

```
## Not run:
##-- Continuing the ITE(.) example:
sEst = summary(Est)
sEst

## End(Not run)
```

---

summary.LF *Summarizing LF*

Description

'summary' method for class 'LF'

Usage

```
## S3 method for class 'LF'
summary(obj)
```

Arguments

obj	An object of class 'LF', a result of a call to 'LF'

Value

The function 'summary.LF' computes and returns a list of summary statistics of LF given 'obj'

output.est	a $ncol(loading.mat)$ x 7 matrix with columns for the loading, plugin(biased) estimators, bias-corrected estimators, its standard error, z-statistic, corresponding (two-sided) p-value and significance stars; Each row corresponds to each loading.

Examples

```
## Not run:
##-- Continuing the LF(.) example:
sEst = summary(Est)
sEst

## End(Not run)
```

---

summary.QF *Summarizing QF*

Description

'summary' method for class 'QF'

Usage

```
## S3 method for class 'QF'
summary(obj)
```

Arguments

obj	An object of class 'QF', a result of a call to 'QF'

Value

The function 'summary.QF' computes and returns a list of summary statistics of LF given 'obj'

output.est	a $length(tau.vec)$ x 7 matrix with columns for tau, plugin(biased) estimators, bias-corrected estimators, its standard error, z-statistic, corresponding (two-sided) p-value and significance stars; Each row corresponds to each tau.

Examples

```
## Not run:
##-- Continuing the QF(.) example:
sEst = summary(Est)
sEst

## End(Not run)
```

Index