Package 'DDL'

January 20, 2025

Type Package

Title Doubly Debiased Lasso (DDL)

Version 1.0.2

Description
Statistical inference for the regression coefficients in high-dimensional linear models with hidden confounders. The Doubly Debiased Lasso method was proposed in <arXiv:2004.03758>.

License MIT + file LICENSE

Encoding UTF-8

Imports stats, glmnet, Matrix

RoxygenNote 7.1.2

NeedsCompilation no

Author Domagoj Ćevid [aut],
Chengzhu Huang [aut],
Zijian Guo [aut, cre],
Peter Bühlmann [aut]

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

Repository CRAN

Date/Publication 2023-04-09 15:20:01 UTC

Contents

ci *Computing confidence intervals*

Description

generic function

Usage

```
ci(x, alpha = 0.05, alternative = c("two.sided", "less", "greater"))
```

Arguments

x	An object of class
alpha	alpha Level of significance to construct confidence interval
alternative	indicates the alternative hypothesis to construct confidence interval and must be one of "two.sided" (default), "less", or "greater".

Examples

```
index = 1
n=100
p=200
s=5
q=3
sigmaE=2
sigma=2
pert=1

H = pert*matrix(rnorm(n*q,mean=0,sd=1),n,q,byrow = TRUE)
Gamma = matrix(rnorm(q*p,mean=0,sd=1),q,p,byrow = TRUE)
#value of X independent from H
E = matrix(rnorm(n*p,mean=0,sd=sigmaE),n,p,byrow = TRUE)

#defined in eq. (2), high-dimensional measured covariates
X = E + H %*% Gamma

delta = matrix(rnorm(q*1,mean=0,sd=1),q,1,byrow = TRUE)

#px1 matrix, creates beta with 1s in the first s entries and the remaining p-s as 0s
beta = matrix(rep(c(1,0),times = c(s,p-s)),p,1,byrow = TRUE)

#nx1 matrix with values of mean 0 and SD of sigma, error in Y independent of X
nu = matrix(rnorm(n*1,mean=0,sd=sigma),n,1,byrow = TRUE)

#eq. (1), the response of the Structural Equation Model
Y = X %*% beta + H %*% delta + nu

result = DDL(X, Y, index)
```

```
# default alpha is 0.05
ci(result, alpha = 0.05)
ci(result, alpha = 0.05, alternative = "less")
ci(result, alpha = 0.05, alternative = "greater")
```

---

ci.DDL *Computing confidence intervals*

Description

'ci' method for class 'DDL'

Usage

```
## S3 method for class 'DDL'
ci(x, alpha = 0.05, alternative = c("two.sided", "less", "greater"))
```

Arguments

x	An object of class 'DDL'
alpha	alpha Level of significance to construct confidence interval
alternative	indicates the alternative hypothesis to construct confidence interval and must be one of "two.sided" (default), "less", or "greater".

Examples

```
index = 1
n=100
p=200
s=5
q=3
sigmaE=2
sigma=2
pert=1

H = pert*matrix(rnorm(n*q,mean=0,sd=1),n,q,byrow = TRUE)
Gamma = matrix(rnorm(q*p,mean=0,sd=1),q,p,byrow = TRUE)
#value of X independent from H
E = matrix(rnorm(n*p,mean=0,sd=sigmaE),n,p,byrow = TRUE)

#defined in eq. (2), high-dimensional measured covariates
X = E + H %*% Gamma

delta = matrix(rnorm(q*1,mean=0,sd=1),q,1,byrow = TRUE)

#px1 matrix, creates beta with 1s in the first s entries and the remaining p-s as 0s
beta = matrix(rep(c(1,0),times = c(s,p-s)),p,1,byrow = TRUE)

#nx1 matrix with values of mean 0 and SD of sigma, error in Y independent of X
```

```
nu = matrix(rnorm(n*1,mean=0,sd=sigma),n,1,byrow = TRUE)

#eq. (1), the response of the Structural Equation Model
Y = X %*% beta + H %*% delta + nu

result = DDL(X, Y, index)
# default alpha is 0.05
ci(result, alpha = 0.05)
ci(result, alpha = 0.05, alternative = "less")
ci(result, alpha = 0.05, alternative = "greater")
```

DDL *Point estimation and inference for a single regression coefficient in the high-dimensional linear model with hidden confounders.*

Description

Computes the Doubly Debiased Lasso estimator of a single regression coefficient in the high-dimensional linear model with hidden confounders. It also constructs the confidence interval for the target regression coefficient.

Usage

```
DDL(X, Y, index, rho = 0.5, rhop = 0.5)
```

Arguments

X	the covariates matrix, of dimension $n \times p$
Y	the outcome vector, of length n
index	the vector of indexes for the regression coefficient of interest
rho	the trim level for X, default is 0.5
rhop	the trim level for X_{-j}, default is 0.5

Value

index	the vector of indexes for the regression coefficient of interest
est_ddl	The vector of the Doubly Debiased Lasso estimator of the target regression coefficient
se	The vector of the standard error of the Doubly Debiased Lasso estimator
est_init	The vector of the spectral deconfounding estimator of the whole regression vector

Examples

```
index = c(1,2,10)
n=100
p=200
s=5
q=3
sigmaE=2
sigma=2
pert=1

H = pert*matrix(rnorm(n*q,mean=0,sd=1),n,q,byrow = TRUE)
Gamma = matrix(rnorm(q*p,mean=0,sd=1),q,p,byrow = TRUE)
#value of X independent from H
E = matrix(rnorm(n*p,mean=0,sd=sigmaE),n,p,byrow = TRUE)

#defined in eq. (2), high-dimensional measured covariates
X = E + H %*% Gamma

delta = matrix(rnorm(q*1,mean=0,sd=1),q,1,byrow = TRUE)

#px1 matrix, creates beta with 1s in the first s entries and the remaining p-s as 0s
beta = matrix(rep(c(1,0),times = c(s,p-s)),p,1,byrow = TRUE)

#nx1 matrix with values of mean 0 and SD of sigma, error in Y independent of X
nu = matrix(rnorm(n*1,mean=0,sd=sigma),n,1,byrow = TRUE)

#eq. (1), the response of the Structural Equation Model
Y = X %*% beta + H %*% delta + nu

result = DDL(X, Y, index)
summary(result)
```

`print.summary.DDL` *Summarizing DDL*

Description

'summary' method for class 'DDL'

Usage

```
## S3 method for class 'summary.DDL'
print(x, ...)
```

Arguments

`x`	An object of class 'summary.DDL'
`...`	Ignored

summary.DDL *Summarizing DDL*

Description

'summary' method for class 'DDL'

Usage

```
## S3 method for class 'DDL'
summary(object, ...)
```

Arguments

object	An object of class 'DDL'
...	Ignored

Value

The function 'summary.DDL' returns a list of summary statistics of DDL given 'DDL'

Examples

```
index = 1
n=100
p=200
s=5
q=3
sigmaE=2
sigma=2
pert=1

H = pert*matrix(rnorm(n*q,mean=0,sd=1),n,q,byrow = TRUE)
Gamma = matrix(rnorm(q*p,mean=0,sd=1),q,p,byrow = TRUE)
#value of X independent from H
E = matrix(rnorm(n*p,mean=0,sd=sigmaE),n,p,byrow = TRUE)

#defined in eq. (2), high-dimensional measured covariates
X = E + H %*% Gamma

delta = matrix(rnorm(q*1,mean=0,sd=1),q,1,byrow = TRUE)

#px1 matrix, creates beta with 1s in the first s entries and the remaining p-s as 0s
beta = matrix(rep(c(1,0),times = c(s,p-s)),p,1,byrow = TRUE)

#nx1 matrix with values of mean 0 and SD of sigma, error in Y independent of X
nu = matrix(rnorm(n*1,mean=0,sd=sigma),n,1,byrow = TRUE)

#eq. (1), the response of the Structural Equation Model
```

```
Y = X %*% beta + H %*% delta + nu

result = DDL(X, Y, index)
summary(result)
```

Index