Package 'subDebiased'

January 25, 2021

Title Sharp Inference on Selected Subgroup in Observational Studies

Version 0.0.0.9000

Author Waverly(Linqing) Wei <linqing_wei@berkeley.edu>

Description This package implements bootstrap-assisted desparsified Lasso and bootstrap-assisted R-split estimators on selected subgroup's treatment effect estimation. The implemented estimators remove the subgroup selection bias and the regularization bias induced by high-dimensional covariates.

License GPL-2

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports snow,parallel,rlecuyer,MASS,glmnet,hdi,foreach,doParallel,doSNOW

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

BSciCoverfun *Compute CI for bootstrap-calibrated methods*

Description

Compute CI for bootstrap-calibrated methods

Usage

```
BSciCoverfun(beta, TB = NULL, G = NULL, alpha = 0.95)
```

Arguments

beta	estimated betas
TB	recalibrated bootstrap statistics
G	indices of subgroups
alpha:	confidence level

Value

LowerBound	Lower confidence bound
UpperBound	Upper confidence bound
betaMax	debiased maximum beta estimate

BSDesparseLasso *Bootstrap-calibrated Desparsified Lasso*

Description

This method first constructs the debiased estimator of β via the desparsified Lasso procedure. Then it calculates the calibration term $\hat{b}_{max} = (1 - n^{r-0.5})(\hat{\beta}_{max} - \hat{\beta}_{j,lasso})$. Through B bootstrap iterations, it recalibrates the bootstrap statistic T_b. The bias-reduced estimate is computed as: $\hat{b}_{max} - \frac{1}{B}\sum_{b=1}^{B} T_b$.

Usage

```
BSDesparseLasso(y, x, r = NULL, G = NULL, B = NULL, alpha = 0.95, fold = 3)
```

Arguments

y	response
x	design matrix
r	tuning parameter
G	subgroup indicator
B	bootstrap iterations
alpha	level of CI

Value

LowerBound	lower confidence bound
UpperBound	upper confidence bound
betaMax	bias-reduced maximum beta estimate
betaEst	debiased beta estimate for each subgroup
op	optimal tuning

BSSplitLasso *Bootstrap-calibrated R-split method*

Description

This method first obtains the estimate of β via repetitive splitting procedure (R-Split) through BB iterations. Then it calculates the calibration term $\tilde{b}_{max} = (1 - n^{r-0.5})(\tilde{\beta}_{max} - \tilde{\beta}_j)$. Through B iterations, it recalibrates the bootstrap statistic T_b. The bias-reduced estimate is computed as: $\tilde{b}_{max} - \frac{1}{B}\sum_{b=1}^{B} T_b$.

Usage

```
BSSplitLasso(
  y,
  x,
  r = NULL,
  G = NULL,
  B = NULL,
  BB = NULL,
  alpha = 0.95,
  splitRatio = 0.6,
  fold = 2
)
```

Arguments

y	response
x	design matrix
r	tuning parameter
G	subgroup indicator
B	bootstrap number
BB	split number
alpha	level ## change other places
splitRatio	split ratio
fold	cross validation fold

Value

LowerBound	lower confidence bound
UpperBound	upper confidence bound
betaMax	bias-reduced maximum beta estimate
betaEst	debiased beta estimate for each subgroup
op	optimal tuning

---

cvDesparse — *Select the optimal tuning for bootstrap-calibrated desparsified Lasso*

Description

Select the optimal tuning for bootstrap-calibrated desparsified Lasso

Usage

```
cvDesparse(y, x, r = NULL, G = NULL, B = NULL, fold = 3)
```

Arguments

y	response
x	design matrix
r	candidate tuning parameters
G	indices of subgroups
B	bootstrap repetitions
fold	number of folds in cross-validation

Value

op	optimal tuning parameter

---

cvSplit — *Select the optimal tuning for bootstrap-calibrated R-Split through cross-validation*

Description

Select the optimal tuning for bootstrap-calibrated R-Split through cross-validation

Usage

```
cvSplit(y, x, r = NULL, G = NULL, B = NULL, BB = NULL, ratio = NULL, fold = 2)
```

Arguments

y	response
x	design matrix
r	candidate tuning parameter
G	subgroup indicator
B	bootstrap iterations
BB	bootstrap iterations for repetitive splitting
ratio	ratio of data splitting
fold	number of folds in cross-validation

Value

op	optimal tuning parameter

IFvarestbiascorr *Cross-validation metric*

Description

Cross-validation metric

Usage

```
IFvarestbiascorr(Ycount, alphaEst, n = NULL, splitSize = NULL)
```

Arguments

Ycount	Y
alphaEst	estimated values
n	sample size
splitSize	size of each split

Value

mean squared error

sigmaMatNew *Generate different types of covariance matrices*

Description

Generate different types of covariance matrices

Usage

```
sigmaMatNew(p, type = NULL)
```

Arguments

p	dimension of confounders
type	type of matrix

Value

Sigma	A covariance matrix

Zmatrix *Generate the nodewise Lasso matrix used in desparsified Lasso*

Description

Generate the nodewise Lasso matrix used in desparsified Lasso

Usage

```
Zmatrix(x, G = NULL)
```

Arguments

x	nodewise confounder matrix
G	indices of subgroups

Value

Z	nodewise Lasso matrix

Index