Package ‘SIRthresholded’

July 21, 2025

Type Package

Title Sliced Inverse Regression with Thresholding

Version 1.0.2

Author Clement Weinreich [aut, cre],
Jerome Saracco [aut],
Hadrien Lorenzo [aut]

Maintainer Clement Weinreich `<clement@weinreich.fr>`

License GPL (>= 2.0)

Description Implements a thresholded version of the Sliced Inverse Regression method (Li, K. C. (1991) <doi:10.2307/2290563>), which allows to do variable selection.

Encoding UTF-8

RoxygenNote 7.2.0

Imports strucchange

Suggests knitr, rmarkdown, mvtnorm

VignetteBuilder knitr

URL `https://clement-w.github.io/SIRthresholded/`

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-09 07:32:54 UTC

Contents

plot.SIR 2
plot.SIR_bootstrap 3
plot.SIR_threshold 4
plot.SIR_threshold_bootstrap 5
plot.SIR_threshold_opt 6
SIR 7
SIR_bootstrap 8
SIR_threshold 9
SIR_threshold_bootstrap 11
SIR_threshold_opt 13

Index **16**

`plot.SIR` *Graphical output of SIR*

Description

Display the 10 first eigen values and the estimated index versus Y of the SIR model.

Usage

```
## S3 method for class 'SIR'
plot(x, choice = "", ...)
```

Arguments

`x`	A SIR object
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs (default).
`...`	arguments to be passed to methods, such as graphical parameters (not used here).

Value

No return value

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR
res = SIR(Y, X, H = 10, graph = FALSE)

# Eigen values
plot(res,choice="eigvals")

# Estimated index versus Y
plot(res,choice="estim_ind")
```

`plot.SIR_bootstrap` *Graphical output of SIR_bootstrap*

Description

Display the 10 first eigen values and the estimated index versus Y of the SIRbootstrap model.

Usage

```
## S3 method for class 'SIR_bootstrap'
plot(x, choice = "", ...)
```

Arguments

`x`	A SIR_bootstrap object
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs (default).
`...`	arguments to be passed to methods, such as graphical parameters (not used here).

Value

No return value

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply bootstrap SIR
res = SIR_bootstrap(Y, X, H = 10, B = 10)

# Eigen values
plot(res,choice=”eigvals”)

# Estimated index versus Y
plot(res,choice=”estim_ind”)
```

`plot.SIR_threshold` *Graphical output of SIR_threshold*

Description

Display the 10 first eigen values and the estimated index versus Y of the thresholded SIR model.

Usage

```
## S3 method for class 'SIR_threshold'
plot(x, choice = "", ...)
```

Arguments

`x`	A SIR_threshold object
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs (default).
`...`	arguments to be passed to methods, such as graphical parameters (not used here).

Value

No return value

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR with hard thresholding
res = SIR_threshold(Y, X, H = 10, lambda = 0.2, thresholding = "hard")

# Eigen values
plot(res,choice="eigvals")

# Estimated index versus Y
plot(res,choice="estim_ind")
```

`plot.SIR_threshold_bootstrap`

Graphical output of SIR_threshold_bootstrap

Description

Display the estimated index versus Y of the SIR model, the size of the models, the occurrence of variable selection, the distribution of the coefficients of and $\hat{b}$ and the distribution of λ_{opt} found across the replications.

Usage

```
## S3 method for class 'SIR_threshold_bootstrap'
plot(x, choice = "", ...)
```

Arguments

`x`	A SIR_threshold_bootstrap object
`choice`	the graph to plot: • "estim_ind" Plot the estimated index by the SIR model versus Y. • "size" Plot the size of the models across the replications. • "selec_var" Plot the occurrence of the selected variables across the replications. • "coefs_b" Plot the value of $\hat{b}$ across the replications. • "lambdas_replic" Plot the distribution of λ_{opt} across the replications. • "" Plot every graphs (default).
`...`	arguments to be passed to methods, such as graphical parameters (not used here).

Value

No return value

Examples

```
# Generate Data
set.seed(10)
n <- 200
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps


res = SIR_threshold_bootstrap(Y,X,H=10,n_lambda=300,thresholding="hard", n_replications=30,k=2)

# Estimated index versus Y
plot(res,choice="estim_ind")
```

```
# Model size
plot(res,choice="size")

# Selected variables
plot(res,choice="selec_var")

# Coefficients of b
plot(res,choice="coefs_b")

# Optimal lambdas
plot(res,choice="lambdas_replic")
```

`plot.SIR_threshold_opt` *Graphical output of SIR_threshold_opt*

Description

Display the 10 first eigen values,the estimated index versus Y of the SIR model, the evolution of cos^2 and variable selection according to λ, and the regularization path of $\hat{b}$.

Usage

```
## S3 method for class 'SIR_threshold_opt'
plot(x, choice = "", ...)
```

Arguments

`x`	A SIR_threshold_opt object
`choice`	the graph to plot: • "estim_ind" Plot the estimated index by the SIR model versus Y. • "opt_lambda" Plot the choice of λ_{opt}. • "cos2_selec" Plot the evolution of cos^2 and variable selection according to λ. • "regul_path" Plot the regularization path of $\hat{b}$. • "" Plot every graphs (default).
`...`	arguments to be passed to methods, such as graphical parameters (not used here).

Value

No return value

Examples

```
# Generate Data
set.seed(10)
n <- 200
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR with soft thresholding
res = SIR_threshold_opt(Y,X,H=10,n_lambda=100,thresholding="soft")

# Estimated index versus Y
plot(res,choice="estim_ind")

# Choice of optimal lambda
plot(res,choice="opt_lambda")

# Evolution of cos^2 and var selection according to lambda
plot(res,choice="cos2_selec")

# Regularization path
plot(res,choice="regul_path")
```

`SIR` *Classic SIR*

Description

Apply a single-index SIR on (X,Y) with H slices. This function allows to obtain an estimate of a basis of the EDR (Effective Dimension Reduction) space via the eigenvector $\hat{b}$ associated with the largest nonzero eigenvalue of the matrix of interest $\widehat{\Sigma}_n^{-1}\widehat{\Gamma}_n$. Thus, $\hat{b}$ is an EDR direction.

Usage

```
SIR(Y, X, H = 10, graph = TRUE, choice = "")
```

Arguments

`Y`	A numeric vector representing the dependent variable (a response vector).
`X`	A matrix representing the quantitative explanatory variables (bind by column).
`H`	The chosen number of slices (default is 10).
`graph`	A boolean that must be set to true to display graphics (default is TRUE).
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs. (default)

Value

An object of class SIR, with attributes:

`b`	This is an estimated EDR direction, which is the principal eigenvector of the interest matrix.
`M1`	The interest matrix.
`eig_val`	The eigenvalues of the interest matrix.
`n`	Sample size.
`p`	The number of variables in X.
`H`	The chosen number of slices.
`call`	Unevaluated call to the function.
`index_pred`	The index Xb' estimated by SIR.
`Y`	The response vector.

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR
SIR(Y, X, H = 10)
```

`SIR_bootstrap` *Bootstrap SIR*

Description

Apply a single-index SIR on B bootstraped samples of (X, Y) with H slices.

Usage

```
SIR_bootstrap(Y, X, H = 10, B = 10, graph = TRUE, choice = "")
```

Arguments

`Y`	A numeric vector representing the dependent variable (a response vector).
`X`	A matrix representing the quantitative explanatory variables (bind by column).
`H`	The chosen number of slices (default is 10).
`B`	The number of bootstrapped samples to draw (default is 10).

`graph`	A boolean that must be set to true to display graphics (default is TRUE).
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs (default).

Value

An object of class SIR_bootstrap, with attributes:

`b`	This is an estimated EDR direction, which is the principal eigenvector of the interest matrix.
`mat_b`	A matrix of size p*B that contains an estimation of beta in the columns for each bootstrapped sample.
`n`	Sample size.
`p`	The number of variables in X.
`H`	The chosen number of slices.
`call`	Unevaluated call to the function.
`index_pred`	The index b'X estimated by SIR.
`Y`	The response vector.

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply bootstrap SIR
SIR_bootstrap(Y, X, H = 10, B = 10)
```

`SIR_threshold` *SIR threshold*

Description

Apply a single-index SIR on (X,Y) with H slices, with a parameter λ which apply a soft/hard thresholding to the interest matrix $\widehat{\Sigma}_n^{-1}\widehat{\Gamma}_n$.

Usage

```
SIR_threshold(
  Y,
  X,
  H = 10,
  lambda = 0,
  thresholding = "hard",
  graph = TRUE,
  choice = ""
)
```

Arguments

`Y`	A numeric vector representing the dependent variable (a response vector).
`X`	A matrix representing the quantitative explanatory variables (bind by column).
`H`	The chosen number of slices (default is 10).
`lambda`	The thresholding parameter (default is 0).
`thresholding`	The thresholding method to choose between hard and soft (default is hard).
`graph`	A boolean that must be set to true to display graphics (default is TRUE).
`choice`	the graph to plot: • "eigvals" Plot the eigen values of the matrix of interest. • "estim_ind" Plot the estimated index by the SIR model versus Y. • "" Plot every graphs (default).

Value

An object of class SIR_threshold, with attributes:

`b`	This is an estimated EDR direction, which is the principal eigenvector of the interest matrix.
`M1`	The interest matrix thresholded.
`eig_val`	The eigenvalues of the interest matrix thresholded.
`eig_vect`	A matrix corresponding to the eigenvectors of the interest matrix.
`Y`	The response vector.
`n`	Sample size.
`p`	The number of variables in X.
`H`	The chosen number of slices.
`nb.zeros`	The number of 0 in the estimation of the vector beta.
`index_pred`	The index Xb' estimated by SIR.
`list.relevant.variables`	A list that contains the variables selected by the model.
`cos_squared`	The cosine squared between vanilla SIR and SIR thresholded.
`lambda`	The thresholding parameter used.

`thresholding`	The thresholding method used.
`call`	Unevaluated call to the function.
`X_reduced`	The X data restricted to the variables selected by the model. It can be used to estimate a new SIR model on the relevant variables to improve the estimation of b.

Examples

```
# Generate Data
set.seed(10)
n <- 500
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR with hard thresholding
SIR_threshold(Y, X, H = 10, lambda = 0.2, thresholding = "hard")
```

`SIR_threshold_bootstrap`

SIR optimally thresholded on bootstraped replications

Description

Apply a single-index optimally soft/hard thresholded SIR with H slices on ‘n_replications‘ bootstraped replications of (X, Y). The optimal number of selected variables is the number of selected variables that came back most often among the replications performed. From this, we can get the corresponding $\hat{b}$ and λ_{opt} that produce the same number of selected variables in the result of ‘SIR_threshold_opt‘.

Usage

```
SIR_threshold_bootstrap(
  Y,
  X,
  H = 10,
  thresholding = "hard",
  n_replications = 50,
  graph = TRUE,
  output = TRUE,
  n_lambda = 100,
  k = 2,
  choice = ""
)
```

Arguments

`Y`	A numeric vector representing the dependent variable (a response vector).
`X`	A matrix representing the quantitative explanatory variables (bind by column).
`H`	The chosen number of slices (default is 10).
`thresholding`	The thresholding method to choose between hard and soft (default is hard).
`n_replications`	The number of bootstraped replications of (X,Y) done to estimate the model (default is 50).
`graph`	A boolean, set to TRUE to plot graphs (default is TRUE).
`output`	A boolean, set to TRUE to print information (default is TRUE).
`n_lambda`	The number of lambda to test. The n_lambda tested lambdas are uniformally distributed between 0 and the maximum value of the interest matrix (default is 100).
`k`	Multiplication factor of the bootstrapped sample size (default is 1 = keep the same size as original data).
`choice`	the graph to plot: • "estim_ind" Plot the estimated index by the SIR model versus Y. • "size" Plot the size of the models across the replications. • "selec_var" Plot the occurrence of the selected variables across the replications. • "coefs_b" Plot the value of b across the replications. • "lambdas_replic" Plot the optimal lambdas across the replications. • "" Plot every graphs (default).

Value

An object of class SIR_threshold_bootstrap, with attributes:

`b`	This is the optimal estimated EDR direction, which is the principal eigenvector of the interest matrix.
`lambda_opt`	The optimal lambda.
`vec_nb_var_selec`	Vector that contains the number of selected variables for each replications.
`occurrences_var`	Vector that contains at index i the number of times the i_th variable has been selected in a replication.
`call`	Unevaluated call to the function.
`nb_var_selec_opt`	Optimal number of selected variables which is the number of selected variables that came back most often among the replications performed.
`list_relevant_variables`	A list that contains the variables selected by the model.
`n`	Sample size.
`p`	The number of variables in X.

`H`	The chosen number of slices.
`n_replications`	The number of bootstraped replications of (X,Y) done to estimate the model.
`thresholding`	The thresholding method used.
`X_reduced`	The X data restricted to the variables selected by the model. It can be used to estimate a new SIR model on the relevant variables to improve the estimation of b.
`mat_b`	Contains the estimation b at each bootstraped replications.
`lambdas_opt_boot`	Contains the optimal lambda found by SIR_threshold_opt at each replication.
`index_pred`	The index Xb' estimated by SIR.
`Y`	The response vector.
`M1`	The interest matrix thresholded with the optimal lambda.

Examples

```
# Generate Data
set.seed(8)
n <-  170
beta <- c(1,1,1,1,1,rep(0,15))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,20))
eps <- rnorm(n,sd=8)
Y <- (X%*%beta)**3+eps

# Apply SIR with hard thresholding
SIR_threshold_bootstrap(Y,X,H=10,n_lambda=300,thresholding="hard", n_replications=30,k=2)
```

`SIR_threshold_opt` *SIR optimally thresholded*

Description

Apply a single-index SIR on (X, Y) with H slices, with a soft/hard thresholding of the interest matrix $\widehat{\Sigma}_n^{-1}\widehat{\Gamma}_n$ by an optimal parameter λ_{opt}. The λ_{opt} is found automatically among a vector of `n_lambda` λ, starting from 0 to the maximum value of $\widehat{\Sigma}_n^{-1}\widehat{\Gamma}_n$. For each feature of X, the number of λ associated with a selection of this feature is stored (in a vector of size p). This vector is sorted in a decreasing way. Then, thanks to `strucchange::breakpoints`, a breakpoint is found in this sorted vector. The coefficients of the variables at the left of the breakpoint, tend to be automatically toggled to 0 due to the thresholding operation based on λ_{opt}, and so should be removed (useless variables). Finally, λ_{opt} corresponds to the first λ such that the associated $\hat{b}$ provides the same number of zeros as the breakpoint's value.

For example, for $X \in R^{10}$ and `n_lambda=100`, this sorted vector can look like this :

X10	X3	X8	X5	X7	X9	X4	X6	X2	X1

2 3 3 4 4 4 6 10 95 100

Here, the breakpoint would be 8.

Usage

```
SIR_threshold_opt(
  Y,
  X,
  H = 10,
  n_lambda = 100,
  thresholding = "hard",
  graph = TRUE,
  output = TRUE,
  choice = ""
)
```

Arguments

`Y`	A numeric vector representing the dependent variable (a response vector).
`X`	A matrix representing the quantitative explanatory variables (bind by column).
`H`	The chosen number of slices (default is 10).
`n_lambda`	The number of lambda to test. The n_lambda tested lambdas are uniformally distributed between 0 and the maximum value of the interest matrix. (default is 100).
`thresholding`	The thresholding method to choose between hard and soft (default is hard).
`graph`	A boolean, set to TRUE to plot graphs (default is TRUE).
`output`	A boolean, set to TRUE to print informations (default is TRUE).
`choice`	the graph to plot: • "estim_ind" Plot the estimated index by the SIR model versus Y. • "opt_lambda" Plot the choice of the optimal lambda. • "cos2_selec" Plot the evolution of cos^2 and variable selection according to lambda. • "regul_path" Plot the regularization path of b. • "" Plot every graphs (default).

Value

An object of class SIR_threshold_opt, with attributes:

`b`	This is the optimal estimated EDR direction, which is the principal eigenvector of the interest matrix.
`lambdas`	A vector that contains the tested lambdas.
`lambda_opt`	The optimal lambda.

`mat_b`	A matrix of size p*n_lambda that contains an estimation of beta in the columns for each lambda.
`n_lambda`	The number of lambda tested.
`vect_nb_zeros`	The number of 0 in b for each lambda.
`list_relevant_variables`	A list that contains the variables selected by the model.
`fit_bp`	An object of class breakpoints from the strucchange package, that contains informations about the breakpoint which allows to deduce the optimal lambda.
`indices_useless_var`	A vector that contains p items: each variable is associated with the number of lambda that selects this variable.
`vect_cos_squared`	A vector that contains for each lambda, the cosine squared between vanilla SIR and SIR thresholded.
`Y`	The response vector.
`n`	Sample size.
`p`	The number of variables in X.
`H`	The chosen number of slices.
`M1`	The interest matrix thresholded with the optimal lambda.
`thresholding`	The thresholding method used.
`call`	Unevaluated call to the function.
`X_reduced`	The X data restricted to the variables selected by the model. It can be used to estimate a new SIR model on the relevant variables to improve the estimation of b.
`index_pred`	The index Xb' estimated by SIR.

Examples

```
# Generate Data
set.seed(2)
n <- 200
beta <- c(1,1,rep(0,8))
X <- mvtnorm::rmvnorm(n,sigma=diag(1,10))
eps <- rnorm(n)
Y <- (X%*%beta)**3+eps

# Apply SIR with soft thresholding
SIR_threshold_opt(Y,X,H=10,n_lambda=300,thresholding="soft")
```

Index

plot.SIR, 2
plot.SIR_bootstrap, 3
plot.SIR_threshold, 4
plot.SIR_threshold_bootstrap, 5
plot.SIR_threshold_opt, 6

SIR, 7
SIR_bootstrap, 8
SIR_threshold, 9
SIR_threshold_bootstrap, 11
SIR_threshold_opt, 13