Package 'TRexSelector'

February 24, 2024

Title T-Rex Selector: High-Dimensional Variable Selection & FDR Control

Version 1.0.0

Date 2024-02-23

Description Performs fast variable selection in high-dimensional settings while controlling the false discovery rate (FDR) at a user-defined target level. The package is based on the paper Machkour, Muma, and Palomar (2022) <arXiv:2110.06048>.

Maintainer Jasin Machkour <jasin.machkour@tu-darmstadt.de>

URL https://github.com/jasinmachkour/TRexSelector, https://arxiv.org/abs/2110.06048

BugReports https://github.com/jasinmachkour/TRexSelector/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

biocViews

Suggests knitr, rmarkdown, ggplot2, patchwork, WGCNA, fastcluster, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports MASS, stats, tlars, parallel, doParallel, foreach, doRNG, methods, glmnet, boot

Depends R (>= 2.10)

VignetteBuilder knitr

NeedsCompilation no

Author Jasin Machkour [aut, cre], Simon Tien [aut], Daniel P. Palomar [aut], Michael Muma [aut]

Repository CRAN

Date/Publication 2024-02-23 23:20:02 UTC

R topics documented:

`add_dummies` *Add dummy predictors to the original predictor matrix*

Description

Sample num_dummies dummy predictors from the univariate standard normal distribution and append them to the predictor matrix X.

Usage

```
add_dummies(X, num_dummies)
```

Arguments

`X`	Real valued predictor matrix.
`num_dummies`	Number of dummies that are appended to the predictor matrix.

Value

Enlarged predictor matrix.

Examples

```
set.seed(123)
n <- 50
p <- 100
X <- matrix(stats::rnorm(n * p), nrow = n, ncol = p)
add_dummies(X = X, num_dummies = p)
```

`add_dummies_GVS` *Add dummy predictors to the original predictor matrix, as required by the T-Rex+GVS selector (*Rhref*https://doi.org/10.23919/EUSIPCO55093.2022.9909883doi:10.23919/EUSIPCO55093.2022.9909883)*

Description

Generate num_dummies dummy predictors as required for the T-Rex+GVS selector (doi:10.23919/EUSIPCO55093.2022.9909883) and append them to the predictor matrix X.

Usage

```
add_dummies_GVS(X, num_dummies, corr_max = 0.5)
```

Arguments

`X`	Real valued predictor matrix.
`num_dummies`	Number of dummies that are appended to the predictor matrix. Has to be a multiple of the number of original variables.
`corr_max`	Maximum allowed correlation between any two predictors from different clusters.

Value

Enlarged predictor matrix for the T-Rex+GVS selector.

Examples

```
set.seed(123)
n <- 50
p <- 100
X <- matrix(stats::rnorm(n * p), nrow = n, ncol = p)
add_dummies_GVS(X = X, num_dummies = p)
```

`FDP` *False discovery proportion (FDP)*

Description

Computes the FDP based on the estimated and the true regression coefficient vectors.

Usage

```
FDP(beta_hat, beta, eps = .Machine$double.eps)
```

Arguments

beta_hat	Estimated regression coefficient vector.
beta	True regression coefficient vector.
eps	Numerical zero.

Value

False discovery proportion (FDP).

Examples

```
data("Gauss_data")
X <- Gauss_data$X
y <- c(Gauss_data$y)
beta <- Gauss_data$beta

set.seed(1234)
res <- trex(X, y)
beta_hat <- res$selected_var

FDP(beta_hat = beta_hat, beta = beta)
```

---

fdp_hat *Computes the conservative FDP estimate of the T-Rex selector (Rhrefhttps://doi.org/10.48550/arXiv.2110.06048doi:10.48550/arXiv.2110.06048)*

Description

Computes the conservative FDP estimate of the T-Rex selector (doi:10.48550/arXiv.2110.06048)

Usage

```
fdp_hat(V, Phi, Phi_prime, eps = .Machine$double.eps)
```

Arguments

V	Voting level grid.
Phi	Vector of relative occurrences.
Phi_prime	Vector of deflated relative occurrences.
eps	Numerical zero.

Value

Vector of conservative FDP estimates for each value of the voting level grid.

Gauss_data *Toy data generated from a Gaussian linear model*

Description

A data set containing a predictor matrix X with n = 50 observations and p = 100 variables (predictors), and a sparse parameter vector beta with associated support vector.

Usage

```
Gauss_data
```

Format

A list containing a matrix X and vectors y, beta, and support:

X Predictor matrix, n = 50, p = 100.

y Response vector.

beta Parameter vector.

support Support vector.

Examples

```
# Generated as follows:
set.seed(789)
n <- 50
p <- 100
X <- matrix(stats::rnorm(n * p), nrow = n, ncol = p)
beta <- c(rep(5, times = 3), rep(0, times = 97))
support <- beta > 0
y <- X %*% beta + stats::rnorm(n)
Gauss_data <- list(
  X = X,
  y = y,
  beta = beta,
  support = support
)
```

lm_dummy *Perform one random experiment*

Description

Run one random experiment of the T-Rex selector (doi:10.48550/arXiv.2110.06048), i.e., generates dummies, appends them to the predictor matrix, and runs the forward selection algorithm until it is terminated after T_stop dummies have been selected.

Usage

```
lm_dummy(
  X,
  y,
  model_tlars,
  T_stop = 1,
  num_dummies = ncol(X),
  method = "trex",
  GVS_type = "IEN",
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  early_stop = TRUE,
  verbose = TRUE,
  intercept = FALSE,
  standardize = TRUE
)
```

Arguments

X	Real valued predictor matrix.
y	Response vector.
model_tlars	Object of the class tlars_cpp. It contains all state variables of the previous T-LARS step (necessary for warm-starts, i.e., restarting the forward selection process exactly where it was previously terminated).
T_stop	Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.
num_dummies	Number of dummies that are appended to the predictor matrix.
method	'trex' for the T-Rex selector (doi:10.48550/arXiv.2110.06048), 'trex+GVS' for the T-Rex+GVS selector (doi:10.23919/EUSIPCO55093.2022.9909883), 'trex+DA+AR1' for the T-Rex+DA+AR1 selector, 'trex+DA+equi' for the T-Rex+DA+equi selector, 'trex+DA+BT' for the T-Rex+DA+BT selector (doi:10.48550/arXiv.2401.15796), 'trex+DA+NN' for the T-Rex+DA+NN selector (doi:10.48550/arXiv.2401.15139).
GVS_type	'IEN' for the Informed Elastic Net (doi:10.1109/CAMSAP58249.2023.10403489), 'EN' for the ordinary Elastic Net (doi:10.1111/j.14679868.2005.00503.x).
type	'lar' for 'LARS' and 'lasso' for Lasso.
corr_max	Maximum allowed correlation between any two predictors from different clusters.
lambda_2_lars	lambda_2-value for LARS-based Elastic Net.
early_stop	Logical. If TRUE, then the forward selection process is stopped after T_stop dummies have been included. Otherwise the entire solution path is computed.
verbose	Logical. If TRUE progress in computations is shown when performing T-LARS steps on the created model.
intercept	Logical. If TRUE an intercept is included.
standardize	Logical. If TRUE the predictors are standardized and the response is centered.

Value

Object of the class tlars_cpp.

Examples

```
set.seed(123)
eps <- .Machine$double.eps
n <- 75
p <- 100
X <- matrix(stats::rnorm(n * p), nrow = n, ncol = p)
beta <- c(rep(3, times = 3), rep(0, times = 97))
y <- X %*% beta + rnorm(n)
res <- lm_dummy(X = X, y = y, T_stop = 1, num_dummies = 5 * p)
beta_hat <- res$get_beta()[seq(p)]
support <- abs(beta_hat) > eps
support
```

Phi_prime_fun *Computes the Deflated Relative Occurrences*

Description

Computes the vector of deflated relative occurrences for all variables (i.e., j = 1,..., p) and T = T_stop.

Usage

```
Phi_prime_fun(
  p,
  T_stop,
  num_dummies,
  phi_T_mat,
  Phi,
  eps = .Machine$double.eps
)
```

Arguments

p	Number of candidate variables.
T_stop	Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.
num_dummies	Number of dummies
phi_T_mat	Matrix of relative occurrences for all variables (i.e., j = 1,..., p) and for T = 1, ..., T_stop.
Phi	Vector of relative occurrences for all variables (i.e., j = 1,..., p) at T = T_stop.
eps	Numerical zero.

Value

Vector of deflated relative occurrences for all variables (i.e., j = 1,..., p) and T = T_stop.

`random_experiments` *Run K random experiments*

Description

Run K early terminated T-Rex (doi:10.48550/arXiv.2110.06048) random experiments and compute the matrix of relative occurrences for all variables and all numbers of included variables before stopping.

Usage

```
random_experiments(
  X,
  y,
  K = 20,
  T_stop = 1,
  num_dummies = ncol(X),
  method = "trex",
  GVS_type = "EN",
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  early_stop = TRUE,
  lars_state_list,
  verbose = TRUE,
  intercept = FALSE,
  standardize = TRUE,
  dummy_coef = FALSE,
  parallel_process = FALSE,
  parallel_max_cores = min(K, max(1, parallel::detectCores(logical = FALSE))),
  seed = NULL,
  eps = .Machine$double.eps
)
```

Arguments

`X`	Real valued predictor matrix.
`y`	Response vector.
`K`	Number of random experiments.
`T_stop`	Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.
`num_dummies`	Number of dummies that are appended to the predictor matrix.

`method` 'trex' for the T-Rex selector (doi:10.48550/arXiv.2110.06048), 'trex+GVS' for the T-Rex+GVS selector (doi:10.23919/EUSIPCO55093.2022.9909883), 'trex+DA+AR1' for the T-Rex+DA+AR1 selector, 'trex+DA+equi' for the T-Rex+DA+equi selector, 'trex+DA+BT' for the T-Rex+DA+BT selector (doi:10.48550/arXiv.2401.15796), 'trex+DA+NN' for the T-Rex+DA+NN selector (doi:10.48550/arXiv.2401.15139).

`GVS_type` 'IEN' for the Informed Elastic Net (doi:10.1109/CAMSAP58249.2023.10403489), 'EN' for the ordinary Elastic Net (doi:10.1111/j.14679868.2005.00503.x).

`type` 'lar' for 'LARS' and 'lasso' for Lasso.

`corr_max` Maximum allowed correlation between any two predictors from different clusters (for method = 'trex+GVS').

`lambda_2_lars` lambda_2-value for LARS-based Elastic Net.

`early_stop` Logical. If TRUE, then the forward selection process is stopped after T_stop dummies have been included. Otherwise the entire solution path is computed.

`lars_state_list`
If parallel_process = TRUE: List of state variables of the previous T-LARS steps of the K random experiments (necessary for warm-starts, i.e., restarting the forward selection process exactly where it was previously terminated). If parallel_process = FALSE: List of objects of the class tlars_cpp associated with the K random experiments (necessary for warm-starts, i.e., restarting the forward selection process exactly where it was previously terminated).

`verbose` Logical. If TRUE progress in computations is shown.

`intercept` Logical. If TRUE an intercept is included.

`standardize` Logical. If TRUE the predictors are standardized and the response is centered.

`dummy_coef` Logical. If TRUE a matrix containing the terminal dummy coefficient vectors of all K random experiments as rows is returned.

`parallel_process`
Logical. If TRUE random experiments are executed in parallel.

`parallel_max_cores`
Maximum number of cores to be used for parallel processing.

`seed` Seed for random number generator (ignored if parallel_process = FALSE).

`eps` Numerical zero.

Value

List containing the results of the K random experiments.

Examples

```
set.seed(123)
data("Gauss_data")
X <- Gauss_data$X
y <- c(Gauss_data$y)
res <- random_experiments(X = X, y = y)
relative_occurrences_matrix <- res$phi_T_mat
relative_occurrences_matrix
```

screen_trex *Run the Screen-T-Rex selector (*Rhrefhttps://doi.org/10.1109/SSP53291.2023.10207957doi:10.110
SSP53291.2023.10207957)

Description

The Screen-T-Rex selector (doi:10.1109/SSP53291.2023.10207957) performs very fast variable selection in high-dimensional settings while informing the user about the automatically selected false discovery rate (FDR).

Usage

```
screen_trex(
  X,
  y,
  K = 20,
  R = 1000,
  method = "trex",
  bootstrap = FALSE,
  conf_level_grid = seq(0, 1, by = 0.001),
  cor_coef = NA,
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  rho_thr_DA = 0.02,
  parallel_process = FALSE,
  parallel_max_cores = min(K, max(1, parallel::detectCores(logical = FALSE))),
  seed = NULL,
  eps = .Machine$double.eps,
  verbose = TRUE
)
```

Arguments

X	Real valued predictor matrix.
y	Response vector.
K	Number of random experiments.
R	Number of bootstrap resamples.
method	'trex' for the T-Rex selector (doi:10.48550/arXiv.2110.06048), 'trex+GVS' for the T-Rex+GVS selector (doi:10.23919/EUSIPCO55093.2022.9909883), 'trex+DA+AR1' for the T-Rex+DA+AR1 selector, 'trex+DA+equi' for the T-Rex+DA+equi selector.
bootstrap	Logical. If TRUE Screen-T-Rex is carried out with bootstrapping.
conf_level_grid	Confidence level grid for the bootstrap confidence intervals.

cor_coef	AR(1) autocorrelation coefficient for the T-Rex+DA+AR1 selector or equicorrelation coefficient for the T-Rex+DA+equi selector.
type	'lar' for 'LARS' and 'lasso' for Lasso.
corr_max	Maximum allowed correlation between any two predictors from different clusters.
lambda_2_lars	lambda_2-value for LARS-based Elastic Net.
rho_thr_DA	Correlation threshold for the T-Rex+DA+AR1 selector and the T-Rex+DA+equi selector (i.e., method = 'trex+DA+AR1' or 'trex+DA+equi').
parallel_process	Logical. If TRUE random experiments are executed in parallel.
parallel_max_cores	Maximum number of cores to be used for parallel processing.
seed	Seed for random number generator (ignored if parallel_process = FALSE).
eps	Numerical zero.
verbose	Logical. If TRUE progress in computations is shown.

Value

A list containing the estimated support vector, the automatically selected false discovery rate (FDR) and additional information.

Examples

```
data("Gauss_data")
X <- Gauss_data$X
y <- c(Gauss_data$y)
set.seed(123)
res <- screen_trex(X = X, y = y)
selected_var <- res$selected_var
selected_var
```

---

select_var_fun *Compute set of selected variables*

Description

Computes the set of selected variables and returns the estimated support vector for the T-Rex selector (doi:10.48550/arXiv.2110.06048).

Usage

```
select_var_fun(p, tFDR, T_stop, FDP_hat_mat, Phi_mat, V)
```

Arguments

p	Number of candidate variables.
tFDR	Target FDR level (between 0 and 1, i.e., 0% and 100%).
T_stop	Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.
FDP_hat_mat	Matrix whose rows are the vectors of conservative FDP estimates for each value of the voting level grid.
Phi_mat	Matrix of relative occurrences as determined by the T-Rex calibration algorithm.
V	Voting level grid.

Value

Estimated support vector.

---

select_var_fun_DA_BT *Compute set of selected variables for the T-Rex+DA+BT selector T-Rex+DA+BT selector (Rhrefhttps://doi.org/10.48550/arXiv.2401.15796doi:10.48550/arXiv.2401.15796)*

Description

Computes the set of selected variables and returns the estimated support vector for the T-Rex+DA+BT selector (doi:10.48550/arXiv.2401.15796).

Usage

```
select_var_fun_DA_BT(
  p,
  tFDR,
  T_stop,
  FDP_hat_array_BT,
  Phi_array_BT,
  V,
  rho_grid
)
```

Arguments

p	Number of candidate variables.
tFDR	Target FDR level (between 0 and 1, i.e., 0% and 100%).
T_stop	Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.

FDP_hat_array_BT
: Array containing the conservative FDP estimates for all variables (dimension 1), values of the voting level grid (dimension 2), and values of the dendrogram grid (dimension 3).

Phi_array_BT
: Array of relative occurrences as determined by the T-Rex calibration algorithm.

V
: Voting level grid.

rho_grid
: Dendrogram grid.

Value

List containing the estimated support vector, etc.

TPP *True positive proportion (TPP)*

Description

Computes the TPP based on the estimated and the true regression coefficient vectors.

Usage

```
TPP(beta_hat, beta, eps = .Machine$double.eps)
```

Arguments

beta_hat
: Estimated regression coefficient vector.

beta
: True regression coefficient vector.

eps
: Numerical zero.

Value

True positive proportion (TPP).

Examples

```
data("Gauss_data")
X <- Gauss_data$X
y <- c(Gauss_data$y)
beta <- Gauss_data$beta

set.seed(1234)
res <- trex(X, y)
beta_hat <- res$selected_var

TPP(beta_hat = beta_hat, beta = beta)
```

trex *Run the T-Rex selector (*R*hrefhttps://doi.org/10.48550/arXiv.2110.06048doi:10.48550/arXiv.2110.06048)*

Description

The T-Rex selector (doi:10.48550/arXiv.2110.06048) performs fast variable selection in high-dimensional settings while controlling the false discovery rate (FDR) at a user-defined target level.

Usage

```
trex(
  X,
  y,
  tFDR = 0.2,
  K = 20,
  max_num_dummies = 10,
  max_T_stop = TRUE,
  method = "trex",
  GVS_type = "IEN",
  cor_coef = NA,
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  rho_thr_DA = 0.02,
  hc_dist = "single",
  hc_grid_length = min(20, ncol(X)),
  parallel_process = FALSE,
  parallel_max_cores = min(K, max(1, parallel::detectCores(logical = FALSE))),
  seed = NULL,
  eps = .Machine$double.eps,
  verbose = TRUE
)
```

Arguments

X	Real valued predictor matrix.
y	Response vector.
tFDR	Target FDR level (between 0 and 1, i.e., 0% and 100%).
K	Number of random experiments.
max_num_dummies	Integer factor determining the maximum number of dummies as a multiple of the number of original variables p (i.e., num_dummies = max_num_dummies * p).
max_T_stop	If TRUE the maximum number of dummies that can be included before stopping is set to ceiling(n / 2), where n is the number of data points/observations.

`method`	'trex' for the T-Rex selector (doi:10.48550/arXiv.2110.06048), 'trex+GVS' for the T-Rex+GVS selector (doi:10.23919/EUSIPCO55093.2022.9909883), 'trex+DA+AR1' for the T-Rex+DA+AR1 selector, 'trex+DA+equi' for the T-Rex+DA+equi selector, 'trex+DA+BT' for the T-Rex+DA+BT selector (doi:10.48550/arXiv.2401.15796), 'trex+DA+NN' for the T-Rex+DA+NN selector (doi:10.48550/arXiv.2401.15139).
`GVS_type`	'IEN' for the Informed Elastic Net (doi:10.1109/CAMSAP58249.2023.10403489), 'EN' for the ordinary Elastic Net (doi:10.1111/j.14679868.2005.00503.x).
`cor_coef`	AR(1) autocorrelation coefficient for the T-Rex+DA+AR1 selector or equicorrelation coefficient for the T-Rex+DA+equi selector.
`type`	'lar' for 'LARS' and 'lasso' for Lasso.
`corr_max`	Maximum allowed correlation between any two predictors from different clusters (for method = 'trex+GVS').
`lambda_2_lars`	lambda_2-value for LARS-based Elastic Net.
`rho_thr_DA`	Correlation threshold for the T-Rex+DA+AR1 selector and the T-Rex+DA+equi selector (i.e., method = 'trex+DA+AR1' or 'trex+DA+equi').
`hc_dist`	Distance measure of the hierarchical clustering/dendrogram (only for trex+DA+BT): 'single' for single-linkage, "complete" for complete linkage, "average" for average linkage (see hclust for more options).
`hc_grid_length`	Length of the height-cutoff-grid for the dendrogram (integer between 1 and the number of original variables p).
`parallel_process`	Logical. If TRUE random experiments are executed in parallel.
`parallel_max_cores`	Maximum number of cores to be used for parallel processing.
`seed`	Seed for random number generator (ignored if parallel_process = FALSE).
`eps`	Numerical zero.
`verbose`	Logical. If TRUE progress in computations is shown.

Value

A list containing the estimated support vector and additional information, including the number of used dummies and the number of included dummies before stopping.

Examples

```
data("Gauss_data")
X <- Gauss_data$X
y <- c(Gauss_data$y)
set.seed(1234)
res <- trex(X = X, y = y)
selected_var <- res$selected_var
selected_var
```

Index