Package ‘TRexSelector’

August 17, 2022

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

Version 0.0.1

Date 2022-08-15

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 (2021) <arXiv:2110.06048>.

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

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

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

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

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

Config/testthat/edition 3

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

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 2022-08-17 06:50:06 UTC
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
```r
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

Description

Generate num_dummies dummy predictors as required for the T-Rex+GVS selector 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**

```r
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)
```

---

**Description**

Computes the conservative FDP estimate of the T-Rex selector

**Usage**

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

**Arguments**

- V: Voting level grid.
- Phi: Vector of relative occurrences.
- Phi_prime: Vector of deflated relative occurrences.
- T_stop: Number of included dummies after which the random experiments (i.e., forward selection processes) are stopped.
- num_dummies: Number of dummies.
- 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.
- $\text{support}$ Support vector.

**Examples**

```r
# 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, i.e., generates dummies, appends them to the predictor matrix, and runs the forward selection algorithm until it is terminated after $T_{\text{stop}}$ dummies have been selected.
Usage

```
lm_dummy(
  X, 
  y, 
  model_tlars, 
  T_stop = 1, 
  num_dummies = ncol(X), 
  method = "trex", 
  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 and 'trex+GVS' for the T-Rex+GVS 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.
- **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**

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

**Usage**

```r
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**

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

Run K random experiments

Description

Run K 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",
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  early_stop = TRUE,
  lars_state_list,
  verbose = TRUE,
  intercept = FALSE,
  standardize = TRUE,
  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 and 'trex+GVS' for the T-Rex+GVS 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.
select_var_fun

`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.

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

`parallel_max_cores` Maximum number of cores to be used for parallel processing (default: minimumNumber of random experiments K, number of physical cores).

`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
```

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

Description

Computes the set of selected variables and returns the estimated support vector for the T-Rex selector.

Usage

```r
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.

---

**TPP**  
*True positive proportion (TPP)*

Description

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

Usage

```r
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

```r
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)
```
Run the T-Rex selector

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

Usage

```r
trex(
  X,
  y,
  tFDR = 0.2,
  K = 20,
  max_num_dummies = 10,
  max_T_stop = TRUE,
  method = "trex",
  type = "lar",
  corr_max = 0.5,
  lambda_2_lars = NULL,
  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 and 'trex+GVS' for the T-Rex+GVS 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.
parallel_process  Logical. If TRUE random experiments are executed in parallel.
parallel_max_cores  Maximum number of cores to be used for parallel processing (default: minimum Number of random experiments K, number of physical cores).
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

```r
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

∗ datasets
  Gauss_data, 5
add_dummies, 2
add_dummies_GVS, 3
FDP, 3
fdp_hat, 4
Gauss_data, 5
lm_dummy, 5
Phi_prime_fun, 7
random_experiments, 8
select_var_fun, 9
TPP, 10
trex, 11