Package: roseRF (via r-universe)

November 11, 2024

Type Package

Title ROSE Random Forests for Robust Semiparametric Efficient Estimation

Version 0.1.0

Maintainer Elliot H. Young <ey244@cam.ac.uk>

Description ROSE (RObust Semiparametric Efficient) random forests for robust semiparametric efficient estimation in partially parametric models (containing generalised partially linear models). Details can be found in the paper by Young and Shah (2024) <doi:10.48550/arXiv.2410.03471>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports caret (>= 6.0.93), glmnet (>= 4.1.6), keras, mgcv, mlr (>= 2.19.1), ParamHelpers, ranger (>= 0.14.1), grf, rpart, stats, tuneRanger (>= 0.5), xgboost

Depends R (>= 4.2.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

NeedsCompilation no

Author Elliot H. Young [aut, cre], Rajen D. Shah [aut]

Repository CRAN

Date/Publication 2024-10-11 08:00:02 UTC

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libglu1-mesa-dev libgmp3-dev make libgsl0-dev jags libicu-dev libpng-dev libxml2-dev libmpfr-dev libopenmpi-dev libproj-dev python3

Contents

`print.roseforest` *Print for a rose random forest fitted object*

Description

This is a method that prints a useful summary of aspects of a `roseRF` object fitted by the functions `roseRF_...` in `roseRF`.

Usage

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

Arguments

`x`	a fitted `roseRF` object fitted by `roseRF...`.
`...`	additional arguments

Value

Prints output for `roseRF` object

`roseRF_gplm` *ROSE random forest estimator for the generalised partially linear model*

Description

Estimates the parameter of interest θ_0 in the generalised partially linear model

$$g(\mathbb{E}[Y|X,Z]) = X\theta_0 + f_0(Z),$$

for some (strictly increasing, differentiable) link function g, which can be reposed in terms of the 'nuisance functions' $(\mathbb{E}[X|Z], \mathbb{E}[g(\mathbb{E}[Y|X,Z])|Z])$ as

$$g\big(\mathbb{E}[Y|X,Z]\big) - \mathbb{E}[g(\mathbb{E}[Y|X,Z])|Z]\big) = (X - \mathbb{E}[X|Z])\theta_0.$$

Usage

```
roseRF_gplm(
  y_on_xz_formula,
  y_on_xz_learner,
  y_on_xz_pars = list(),
  Gy_on_z_formula,
  Gy_on_z_learner,
  Gy_on_z_pars = list(),
  x_formula,
  x_learner,
  x_pars = list(),
  M1_formula = x_formula,
  M1_learner = x_learner,
  M1_pars = x_pars,
  M2_formula = NA,
  M2_learner = NA,
  M2_pars = list(),
  M3_formula = NA,
  M3_learner = NA,
  M3_pars = list(),
  M4_formula = NA,
  M4_learner = NA,
  M4_pars = list(),
  M5_formula = NA,
  M5_learner = NA,
  M5_pars = list(),
  link = "identity",
  data,
  K = 5,
  S = 1,
  max.depth = 10,
  num.trees = 500,
  min.node.size = max(10, ceiling(0.01 * (K - 1)/K * nrow(data))),
  replace = TRUE,
  sample.fraction = 0.8
)
```

Arguments

`y_on_xz_formula`	a two-sided formula object describing the model for $\mathbb{E}[Y\|X, Z]$ (regressing Y on (X)).
`y_on_xz_learner`	a string specifying the regression method to fit the regression as given by `y_on_xz_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).
`y_on_xz_pars`	a list containing hyperparameters for the `y_on_xz_learner` chosen. Default is an empty list, which performs hyperparameter tuning.

`Gy_on_z_formula`
: a two-sided formula object describing the model for $\mathbb{E}[g(\mathbb{E}[Y|X,Z])|Z]$ (regressing $g(\hat{E}[Y|X,Z])$ on Z).

`Gy_on_z_learner`
: a string specifying the regression method to fit the regression as given by `Gy_on_z_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).

`Gy_on_z_pars`
: a list containing hyperparameters for the `Gy_on_z_learner` chosen. Default is an empty list, which performs hyperparameter tuning.

`x_formula`
: a two-sided formula object describing the model for $\mathbb{E}[X|Z]$.

`x_learner`
: a string specifying the regression method to fit the regression of X on Z as given by `x_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).

`x_pars`
: a list containing hyperparameters for the `x_learner` chosen. Default is an empty list, which performs hyperparameter tuning.

`M1_formula`
: a two-sided formula object for the model $\mathbb{E}[M_1(X)|Z]$. Default is $M_1(X) = X$.

`M1_learner`
: a string specifying the regression method for $\mathbb{E}[M_1(X)|Z]$ estimation.

`M1_pars`
: a list containing hyperparameters for the `M1_learner` chosen.

`M2_formula`
: a two-sided formula object for the model $\mathbb{E}[M_2(X)|Z]$. Default is no formula / regression (i.e. $J = 1$)

`M2_learner`
: a string specifying the regression method for $\mathbb{E}[M_2(X)|Z]$ estimation.

`M2_pars`
: a list containing hyperparameters for the `M2_learner` chosen.

`M3_formula`
: a two-sided formula object for the model $\mathbb{E}[M_3(X)|Z]$. Default is no formula / regression (i.e. $J = 1$).

`M3_learner`
: a string specifying the regression method for $\mathbb{E}[M_3(X)|Z]$ estimation.

`M3_pars`
: a list containing hyperparameters for the `M3_learner` chosen.

`M4_formula`
: a two-sided formula object for the model $\mathbb{E}[M_4(X)|Z]$. Default is no formula / regression (i.e. $J = 1$)

`M4_learner`
: a string specifying the regression method for $\mathbb{E}[M_4(X)|Z]$ estimation.

`M4_pars`
: a list containing hyperparameters for the `M4_learner` chosen.

`M5_formula`
: a two-sided formula object for the model $\mathbb{E}[M_5(X)|Z]$. Default is no formula / regression (i.e. $J = 1$)

`M5_learner`
: a string specifying the regression method for $\mathbb{E}[M_5(X)|Z]$ estimation.

`M5_pars`
: a list containing hyperparameters for the `M5_learner` chosen.

`link`
: link function (g). Options include `identity`, `log`, `sqrt`, `logit`, `probit`. Default is `identity`.

`data`
: a data frame containing the variables for the partially linear model.

`K`
: the number of folds used for K-fold cross-fitting. Default is 5.

`S`
: the number of repeats to mitigate the randomness in the estimator on the sample splits used for K-fold cross-fitting. Default is 5.

`max.depth`
: Maximum depth parameter used for ROSE random forests. Default is 5.

`num.trees`
: Number of trees used for a single ROSE random forest. Default is 50.

`min.node.size` Minimum node size of a leaf in each tree. Default is `max(10,ceiling(0.01 (K-1)/K nrow(data)))`.

`replace` Whether sampling for a single random tree are performed with (bootstrap) or without replacement. Default is `TRUE` (i.e. bootstrap).

`sample.fraction` Proportion of data used for each random tree. Default is 0.8.

Details

The estimator of interest θ_0 solves the estimating equation

$$\sum_i \psi(Y_i, X_i, Z_i; \theta, \hat{\eta}(Z), \hat{w}(Z)) = 0,$$

$$\psi(Y, X, Z; \theta, \eta_0, w) := \sum_{j=1}^{J} w_j(Z)\big(M_j(X) - \mathbb{E}[M_j(X)|Z]\big) g'(\mu(X, Z; \theta, \eta_0))\big(Y - \mu(X, Z; \theta, \eta_0)\big),$$

$$\mu(X, Z; \theta, \eta_0) := g^{-1}\big(\mathbb{E}[g(\mathbb{E}[Y|X, Z])|Z] + (X - \mathbb{E}[X|Z])\theta\big),$$

$$\eta_0 := \big(\mathbb{E}[Y|Z = \cdot], \mathbb{E}[X|Z = \cdot]\big),$$

where $M_1(X), \ldots, M_J(X)$ denotes user-chosen functions of (X) and $w(Z) = \big(w_1(Z), \ldots, w_J(Z)\big)$ denotes weights estimated via ROSE random forests. The default takes $J = 1$ and $M_1(X) = X$; if taking $J \geq 2$ we recommend care in checking the applicability and appropriateness of any additional user-chosen regression tasks.

The parameter of interest θ_0 is estimated using a DML2 / K-fold cross-fitting framework, to allow for arbitrary (faster than $n^{1/4}$-consistent) learners for $\hat{\eta}$ i.e. solving the estimating equation

$$\sum_{k \in [K]} \sum_{i \in I_k} \psi(Y_i, X_i, Z_i; \theta, \hat{\eta}^{(k)}(Z), \hat{w}^{(k)}(Z)) = 0,$$

where $I_1, \ldots, I_K$ denotes a partition of the index set for the datapoints (Y_i, X_i, Z_i), $\hat{\eta}^{(k)}$ denotes an estimator for η_0 trained on the data indexed by I_k^c, and $\hat{w}^{(k)}$ denotes a ROSE random forest (again trained on the data indexed by I_k^c).

Value

A list containing:

`theta` The estimator of θ_0.

`stderror` Huber robust estimate of the standard error of the θ_0-estimator.

`coefficients` Table of θ_0 coefficient estimator, standard error, z-value and p-value.

`roseRF_pliv` *ROSE random forest estimator for the partially linear instrumental variable model*

Description

ROSE random forest estimator for the partially linear instrumental variable model

Usage

```
roseRF_pliv(
  y_formula,
  y_learner,
  y_pars = list(),
  x_formula,
  x_learner,
  x_pars = list(),
  IV1_formula = NA,
  IV1_learner = NA,
  IV1_pars = list(),
  IV2_formula = NA,
  IV2_learner = NA,
  IV2_pars = list(),
  IV3_formula = NA,
  IV3_learner = NA,
  IV3_pars = list(),
  IV4_formula = NA,
  IV4_learner = NA,
  IV4_pars = list(),
  IV5_formula = NA,
  IV5_learner = NA,
  IV5_pars = list(),
  data,
  K = 5,
  S = 1,
  max.depth = 10,
  num.trees = 500,
  min.node.size = max(10, ceiling(0.01 * (K - 1)/K * nrow(data))),
  replace = TRUE,
  sample.fraction = 0.8
)
```

Arguments

`y_formula`	a two-sided formula object describing the regression model for $\mathbb{E}[Y\|Z]$.
`y_learner`	a string specifying the regression method to fit the regression of Y on Z as given by `y_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).

y_pars	a list containing hyperparameters for the y_learner chosen. Default is an empty list, which performs hyperparameter tuning.
x_formula	a two-sided formula object describing the regression model for $\mathbb{E}[X\|Z]$.
x_learner	a string specifying the regression method to fit the regression of X on Z as given by x_formula (e.g. randomforest, xgboost, neuralnet, gam).
x_pars	a list containing hyperparameters for the y_learner chosen. Default is an empty list, which performs hyperparameter tuning.
IV1_formula	a two-sided formula object for the model $\mathbb{E}[V_1(X)\|Z]$.
IV1_learner	a string specifying the regression method for $\mathbb{E}[V_1(X)\|Z]$ estimation.
IV1_pars	a list containing hyperparameters for the IV1_learner chosen.
IV2_formula	a two-sided formula object for the model $\mathbb{E}[V_2\|Z]$. Default is no formula / regression (i.e. $J = 1$)
IV2_learner	a string specifying the regression method for $\mathbb{E}[V_2(X)\|Z]$ estimation.
IV2_pars	a list containing hyperparameters for the IV2_learner chosen.
IV3_formula	a two-sided formula object for the model $\mathbb{E}[V_3(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$).
IV3_learner	a string specifying the regression method for $\mathbb{E}[V_3(X)\|Z]$ estimation.
IV3_pars	a list containing hyperparameters for the IV3_learner chosen.
IV4_formula	a two-sided formula object for the model $\mathbb{E}[V_4(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$)
IV4_learner	a string specifying the regression method for $\mathbb{E}[V_4(X)\|Z]$ estimation.
IV4_pars	a list containing hyperparameters for the IV4_learner chosen.
IV5_formula	a two-sided formula object for the model $\mathbb{E}[V_5(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$)
IV5_learner	a string specifying the regression method for $\mathbb{E}[V_5(X)\|Z]$ estimation.
IV5_pars	a list containing hyperparameters for the IV5_learner chosen.
data	a data frame containing the variables for the partially linear model.
K	the number of folds used for K-fold cross-fitting. Default is 5.
S	the number of repeats to mitigate the randomness in the estimator on the sample splits used for K-fold cross-fitting. Default is 5.
max.depth	Maximum depth parameter used for ROSE random forests. Default is 5.
num.trees	Number of trees used for a single ROSE random forest. Default is 50.
min.node.size	Minimum node size of a leaf in each tree. Default is max(10,ceiling(0.01 (K-1)/K nrow(data))).
replace	Whether sampling for a single random tree are performed with (bootstrap) or without replacement. Default is TRUE (i.e. bootstrap).
sample.fraction	Proportion of data used for each random tree. Default is 0.8.

Value

A list containing:

`theta` The estimator of θ_0.

`stderror` Huber robust estimate of the standard error of the θ_0-estimator.

`coefficients` Table of θ_0 coefficient estimator, standard error, z-value and p-value.

`roseRF_plm` *ROSE random forest estimator for the partially linear model*

Description

Estimates the parameter of interest θ_0 in the partially linear model

$$\mathbb{E}[Y|X,Z] = X\theta_0 + f_0(Z),$$

which can be reposed in terms of the 'nuisance functions' $(\mathbb{E}[Y|X], \mathbb{E}[X|Z])$ as

$$\mathbb{E}[Y|X,Z] - \mathbb{E}[Y|Z] = (X - \mathbb{E}[X|Z])\theta_0.$$

Usage

```
roseRF_plm(
  y_formula,
  y_learner,
  y_pars = list(),
  x_formula,
  x_learner,
  x_pars = list(),
  M1_formula = x_formula,
  M1_learner = x_learner,
  M1_pars = x_pars,
  M2_formula = NA,
  M2_learner = NA,
  M2_pars = list(),
  M3_formula = NA,
  M3_learner = NA,
  M3_pars = list(),
  M4_formula = NA,
  M4_learner = NA,
  M4_pars = list(),
  M5_formula = NA,
  M5_learner = NA,
  M5_pars = list(),
  data,
  K = 5,
  S = 1,
```

```
  max.depth = 10,
  num.trees = 500,
  min.node.size = max(10, ceiling(0.01 * (K - 1)/K * nrow(data))),
  replace = TRUE,
  sample.fraction = 0.8
)
```

Arguments

y_formula	a two-sided formula object describing the model for $\mathbb{E}[Y\|Z]$.
y_learner	a string specifying the regression method to fit the regression of Y on Z as given by y_formula (e.g. randomforest, xgboost, neuralnet, gam).
y_pars	a list containing hyperparameters for the y_learner chosen. Default is an empty list, which performs hyperparameter tuning.
x_formula	a two-sided formula object describing the model for $\mathbb{E}[X\|Z]$.
x_learner	a string specifying the regression method to fit the regression of X on Z as given by x_formula (e.g. randomforest, xgboost, neuralnet, gam).
x_pars	a list containing hyperparameters for the y_learner chosen. Default is an empty list, which performs hyperparameter tuning.
M1_formula	a two-sided formula object for the model $\mathbb{E}[M_1(X)\|Z]$. Default is $M_1(X) = X$.
M1_learner	a string specifying the regression method for $\mathbb{E}[M_1(X)\|Z]$ estimation.
M1_pars	a list containing hyperparameters for the M1_learner chosen.
M2_formula	a two-sided formula object for the model $\mathbb{E}[M_2(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$)
M2_learner	a string specifying the regression method for $\mathbb{E}[M_2(X)\|Z]$ estimation.
M2_pars	a list containing hyperparameters for the M2_learner chosen.
M3_formula	a two-sided formula object for the model $\mathbb{E}[M_3(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$).
M3_learner	a string specifying the regression method for $\mathbb{E}[M_3(X)\|Z]$ estimation.
M3_pars	a list containing hyperparameters for the M3_learner chosen.
M4_formula	a two-sided formula object for the model $\mathbb{E}[M_4(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$)
M4_learner	a string specifying the regression method for $\mathbb{E}[M_4(X)\|Z]$ estimation.
M4_pars	a list containing hyperparameters for the M4_learner chosen.
M5_formula	a two-sided formula object for the model $\mathbb{E}[M_5(X)\|Z]$. Default is no formula / regression (i.e. $J = 1$)
M5_learner	a string specifying the regression method for $\mathbb{E}[M_5(X)\|Z]$ estimation.
M5_pars	a list containing hyperparameters for the M5_learner chosen.
data	a data frame containing the variables for the partially linear model.
K	the number of folds used for K-fold cross-fitting. Default is 5.
S	the number of repeats to mitigate the randomness in the estimator on the sample splits used for K-fold cross-fitting. Default is 5.

`max.depth` Maximum depth parameter used for ROSE random forests. Default is 5.

`num.trees` Number of trees used for a single ROSE random forest. Default is 50.

`min.node.size` Minimum node size of a leaf in each tree. Default is `max(10,ceiling(0.01 (K-1)/K nrow(data)))`.

`replace` Whether sampling for a single random tree are performed with (bootstrap) or without replacement. Default is `TRUE` (i.e. bootstrap).

`sample.fraction` Proportion of data used for each random tree. Default is 0.8.

Details

The estimator of interest θ_0 solves the estimating equation

$$\sum_i \psi(Y_i, X_i, Z_i; \theta, \hat{\eta}(Z), \hat{w}(Z)) = 0,$$

$$\psi(Y, X, Z; \theta, \eta_0, w) := \sum_{j=1}^{J} w_j(Z)\big(M_j(X) - \mathbb{E}[M_j(X)|Z]\big)\Big(\big(Y - \mathbb{E}[Y|Z]\big) - \big(X - \mathbb{E}[X|Z]\big)\theta\Big),$$

$$\eta_0 := \big(\mathbb{E}[Y|Z = \cdot], \mathbb{E}[X|Z = \cdot]\big),$$

where $M_1(X), \ldots, M_J(X)$ denotes user-chosen functions of (X) and $w(Z) = \big(w_1(Z), \ldots, w_J(Z)\big)$ denotes weights estimated via ROSE random forests. The default takes $J = 1$ and $M_1(X) = X$; if taking $J \geq 2$ we recommend care in checking the applicability and appropriateness of any additional user-chosen regression tasks.

The parameter of interest θ_0 is estimated using a DML2 / K-fold cross-fitting framework, to allow for arbitrary (faster than $n^{1/4}$-consistent) learners for $\hat{\eta}$ i.e. solving the estimating equation

$$\sum_{k \in [K]} \sum_{i \in I_k} \psi(Y_i, X_i, Z_i; \theta, \hat{\eta}^{(k)}(Z), \hat{w}^{(k)}(Z)) = 0,$$

where $I_1, \ldots, I_K$ denotes a partition of the index set for the datapoints (Y_i, X_i, Z_i), $\hat{\eta}^{(k)}$ denotes an estimator for η_0 trained on the data indexed by I_k^c, and $\hat{w}^{(k)}$ denotes a ROSE random forest (again trained on the data indexed by I_k^c).

Value

A list containing:

`theta` The estimator of θ_0.

`stderror` Huber robust estimate of the standard error of the θ_0-estimator.

`coefficients` Table of θ_0 coefficient estimator, standard error, z-value and p-value.

summary.roseforest *Summary for a rose random forest fitted object*

Description

Prints a roseRF object fitted by the functions roseRF_... in roseRF.

Usage

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

Arguments

object	a fitted roseRF object fitted by roseRF_....
...	additional arguments

Value

Prints summary output for roseRF object

unweighted_gplm *Unweighted (baseline) estimator for the generalised partially linear model*

Description

Estimates the parameter of interest θ_0 in the generalised partially linear regression model

$$g(\mathbb{E}[Y|X,Z]) = X\theta_0 + f_0(Z),$$

as in roseRF_gplm but without any weights i.e. $J = 1$, $M_1(X) = X$ and $w_1 \equiv 1$.

Usage

```
unweighted_gplm(
  y_on_xz_formula,
  y_on_xz_learner,
  y_on_xz_pars = list(),
  Gy_on_z_formula,
  Gy_on_z_learner,
  Gy_on_z_pars = list(),
  x_formula,
  x_learner,
  x_pars = list(),
  link = "identity",
```

```
  data,
  K = 5,
  S = 1
)
```

Arguments

y_on_xz_formula	a two-sided formula object describing the regression model for $\mathbb{E}[Y\|X, Z]$ (regressing Y on (X, Z)).
y_on_xz_learner	a string specifying the regression method to fit the regression as given by y_on_xz_formula (e.g. randomforest, xgboost, neuralnet, gam).
y_on_xz_pars	a list containing hyperparameters for the y_on_xz_learner chosen. Default is an empty list, which performs hyperparameter tuning.
Gy_on_z_formula	a two-sided formula object describing the regression model for $\mathbb{E}[g(\mathbb{E}[Y\|X, Z])\|Z]$ (regressing $g(\hat{E}[Y\|X, Z])$ on Z).
Gy_on_z_learner	a string specifying the regression method to fit the regression as given by Gy_on_z_formula (e.g. randomforest, xgboost, neuralnet, gam).
Gy_on_z_pars	a list containing hyperparameters for the Gy_on_z_learner chosen. Default is an empty list, which performs hyperparameter tuning.
x_formula	a two-sided formula object describing the regression model for $\mathbb{E}[X\|Z]$.
x_learner	a string specifying the regression method to fit the regression of X on Z as given by x_formula (e.g. randomforest, xgboost, neuralnet, gam).
x_pars	a list containing hyperparameters for the x_learner chosen. Default is an empty list, which performs hyperparameter tuning.
link	link function (g). Options include identity, log, sqrt, logit, probit. Default is identity.
data	a data frame containing the variables for the partially linear model.
K	the number of folds used for K-fold cross-fitting. Default is 5.
S	the number of repeats to mitigate the randomness in the estimator on the sample splits used for K-fold cross-fitting. Default is 5.

Value

A list containing:

theta The estimator of θ_0.

stderror Huber robust estimate of the standard error of the θ_0-estimator.

coefficients Table of θ_0 coefficient estimator, standard error, z-value and p-value.

unweighted_plm *Unweighted (baseline) estimator for the partially linear model*

Description

Estimates the parameter of interest θ_0 in the partially linear regression model

$$\mathbb{E}[Y|X, Z] = X\theta_0 + f_0(Z),$$

as in `roseRF_plm` but without any weights i.e. $J = 1$, $M_1(X) = X$ and $w_1 \equiv 1$.

Usage

```
unweighted_plm(
  y_formula,
  y_learner,
  y_pars = list(),
  x_formula,
  x_learner,
  x_pars = list(),
  data,
  K = 5,
  S = 1
)
```

Arguments

`y_formula`	a two-sided formula object describing the regression model for $\mathbb{E}[Y\|Z]$.
`y_learner`	a string specifying the regression method to fit the regression of Y on Z as given by `y_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).
`y_pars`	a list containing hyperparameters for the `y_learner` chosen. Default is an empty list, which performs hyperparameter tuning.
`x_formula`	a two-sided formula object describing the regression model for $\mathbb{E}[X\|Z]$.
`x_learner`	a string specifying the regression method to fit the regression of X on Z as given by `x_formula` (e.g. `randomforest`, `xgboost`, `neuralnet`, `gam`).
`x_pars`	a list containing hyperparameters for the `y_learner` chosen. Default is an empty list, which performs hyperparameter tuning.
`data`	a data frame containing the variables for the partially linear model.
`K`	the number of folds used for K-fold cross-fitting. Default is 5.
`S`	the number of repeats to mitigate the randomness in the estimator on the sample splits used for K-fold cross-fitting. Default is 5.

Value

A list containing:

`theta` The estimator of θ_0.

`stderror` Huber robust estimate of the standard error of the θ_0-estimator.

`coefficients` Table of θ_0 coefficient estimator, standard error, z-value and p-value.

Index