Package 'hdcate'

December 14, 2022

Type Package

Title Estimation of Conditional Average Treatment Effects with
High-Dimensional Data

Version 0.1.0

Imports KernSmooth, R6, hdm, locpol, caret

Description A two-step double-robust method to estimate the conditional average treatment effects (CATE) with potentially high-dimensional covariate(s). In the first stage, the nuisance functions necessary for identifying CATE are estimated by machine learning methods, allowing the number of covariates to be comparable to or larger than the sample size. The second stage consists of a low-dimensional local linear regression, reducing CATE to a function of the covariate(s) of interest. The CATE estimator implemented in this package not only allows for high-dimensional data, but also has the "double robustness" property: either the model for the propensity score or the models for the conditional means of the potential outcomes are allowed to be misspecified (but not both). This package is based on the paper by Fan et al., ``Estimation of Conditional Average Treatment Effects With High-Dimensional Data'' (2022), Journal of Business & Economic Statistics <doi:10.1080/07350015.2020.1811102>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, xfun, randomForest, dplyr, ggplot2,
ggthemes

VignetteBuilder knitr

NeedsCompilation no

Author Qingliang Fan [aut, cre],
Hengzhao Hong [aut]

Maintainer Qingliang Fan <michaelqfan@cuhk.edu.hk>

Repository CRAN

Date/Publication 2022-12-14 11:50:02 UTC

R **topics documented:**

HDCATE   *High-Dimensional Conditional Average Treatment Effects (HDCATE)*
                                Estimator

Description

Use a two-step procedure to estimate the conditional average treatment effects (CATE) with potentially high-dimensional covariate(s). Run `browseVignettes('hdcate')` to browse the user manual of this package.

Usage

```
HDCATE(data, y_name, d_name, x_formula)
```

Arguments

data	data frame of the observed data
y_name	variable name of the observed outcomes
d_name	variable name of the treatment indicators
x_formula	formula of the covariates

Value

An initialized `HDCATE` model (object), ready for estimation.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# for example, and alternatively, the propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# Example 1: full-sample estimator
# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

# estimate HDCATE function, inference, and plot
HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)

HDCATE.fit(model)
HDCATE.inference(model)
HDCATE.plot(model)


# Example 2: cross-fitting estimator
# change above estimator to cross-fitting mode, 5 folds, for example.
HDCATE.use_cross_fitting(model, k_fold=5)

# estimate HDCATE function, inference, and plot
HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)

HDCATE.fit(model)
HDCATE.inference(model)
HDCATE.plot(model)
```

HDCATE.fit   *Fit the HDCATE function*

Description

Fit the HDCATE function

Usage

```
HDCATE.fit(HDCATE_model, verbose = TRUE)
```

Arguments

HDCATE_model   an object created via HDCATE

verbose   whether the verbose message is displayed, the default is TRUE

Value

None. The `HDCATE_model` is fitted.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)


HDCATE.fit(model)
```

HDCATE.get_sim_data   *Get simulation data*

Description

Get simulation data

Usage

```
HDCATE.get_sim_data(
  n_obs = 500,
  n_var = 100,
  n_rel_var = 4,
  sig_strength_propensity = 0.5,
  sig_strength_outcome = 1,
  intercept = 10
)
```

Arguments

n_obs   Num of observations

n_var   Num of covariates

`n_rel_var`	Num of relevant variables, only the first `n_rel_var` covariates are actually present in the expectation function of potential outcome, and only the last `n_rel_var` covariates are present in the propensity score function.
`sig_strength_propensity`	
	signal strength in propensity score functions
`sig_strength_outcome`	
	signal strength in outcome functions
`intercept`	value of intercept in outcome functions

Value

a data.frame, which is the simulated observed data.

Examples

```
HDCATE.get_sim_data()
HDCATE.get_sim_data(n_obs=50, n_var=4, n_rel_var=2)
```

`HDCATE.inference`	*Construct uniform confidence bands*

Description

Construct uniform confidence bands

Usage

```
HDCATE.inference(
  HDCATE_model,
  sig_level = 0.01,
  n_rep_boot = 1000,
  verbose = FALSE
)
```

Arguments

`HDCATE_model`	an object created via HDCATE
`sig_level`	a (vector of) significant level, such as 0.01, or c(0.01, 0.05, 0.10)
`n_rep_boot`	repeat n times for bootstrap, the default is 1000
`verbose`	whether the verbose message is displayed, the default is FALSE

Value

None. The HDCATE confidence bands are constructed.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)


HDCATE.fit(model)
HDCATE.inference(model)
```

HDCATE.plot   *Plot HDCATE function and the uniform confidence bands*

Description

Plot HDCATE function and the uniform confidence bands

Usage

```
HDCATE.plot(
  HDCATE_model,
  output_pdf = FALSE,
  pdf_name = "hdcate_plot.pdf",
  include_band = TRUE,
  test_side = "both",
  y_axis_min = "auto",
  y_axis_max = "auto",
  display.hdcate = "HDCATEF",
  display.ate = "ATE",
  display.siglevel = "sig_level"
)
```

Arguments

HDCATE_model	an object created via [HDCATE](#)
output_pdf	if TRUE, the plot will be saved as a PDF file, the default is FALSE
pdf_name	file name when output_pdf=TRUE

include_band	if TRUE, plot the uniform confidence bands (need: HDCATE.inference was called before)
test_side	'both', 'left' or 'right', i.e. 2-side test or one-side test
y_axis_min	minimum value of the Y axis to plot in the graph, the default is auto
y_axis_max	maximum value of the Y axis to plot in the graph, the default is auto
display.hdcate	the name of HDCATE function in the legend, the default is 'HDCATEF'
display.ate	the name of average treatment effect in the legend, the default is 'ATE'
display.siglevel	
	the name of the significant level for confidence bands in the legend, the default is 'sig_level'

Value

None. A plot will be shown or saved as PDF.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)

HDCATE.fit(model)
HDCATE.inference(model)
HDCATE.plot(model)
```

HDCATE.set_bw	*Set bandwidth*

Description

Set user-defined bandwidth.

Usage

```
HDCATE.set_bw(model, bandwidth = "default")
```

Arguments

model   an object created via [HDCATE](#)

bandwidth   the value of bandwidth

Value

None.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

# Set user-defined bandwidth, e.g., 0.15.
HDCATE.set_bw(model, 0.15)
```

HDCATE.set_condition_var
                                 Set the conditional variable in CATE

Description

Set the conditional variable in CATE

Usage

```
HDCATE.set_condition_var(
  HDCATE_model,
  name = NA,
  min = NA,
  max = NA,
  step = NA
)
```

Arguments

`HDCATE_model`	an object created via [HDCATE](#)
`name`	name of the conditional variable
`min`	minimum value of the conditional variable for evaluation
`max`	maximum value of the conditional variable for evaluation
`step`	minimum distance between two evaluation points

Value

None. The `HDCATE_model` is ready to fit.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

HDCATE.set_condition_var(model, 'X2', min=-1, max=1, step=0.01)
```

`HDCATE.set_first_stage`

Set user-defined first-stage estimating methods

Description

Set user-defined ML methods (such as random forests, elastic-net, boosting) to run the first-stage estimation.

Usage

```
HDCATE.set_first_stage(
  model,
  fit.treated,
  fit.untreated,
  fit.propensity,
  predict.treated,
  predict.untreated,
  predict.propensity
)
```

Arguments

`model`	an object created via [HDCATE](#)
`fit.treated`	function that accepts a data.frame as the only argument, fits the treated expectation function, and returns a fitted object
`fit.untreated`	function that accepts a data.frame as the only argument, fits the untreated expectation function, and returns a fitted object
`fit.propensity`	function that accepts a data.frame as the only argument, fits the propensity function, and return a fitted object
`predict.treated`	
	function that accepts the returned object of `fit.treated` and a data.frame as arguments, and returns the predicted vector of that data.frame
`predict.untreated`	
	function that accepts the returned object of `fit.untreated` and a data.frame as arguments, and returns the predicted vector that data.frame
`predict.propensity`	
	function that accepts the returned object of `fit.propensity` and a data.frame as arguments, and returns the predicted vector that data.frame

Value

None.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

# manually define a lasso method
my_lasso_fit_exp <- function(df) {
  hdm::rlasso(as.formula(paste0('Y', "~", x_formula)), df)
}
my_lasso_predict_exp <- function(fitted_model, df) {
  predict(fitted_model, df)
}
my_lasso_fit_ps <- function(df) {
  hdm::rlassologit(as.formula(paste0('D', "~", x_formula)), df)
}
my_lasso_predict_ps <- function(fitted_model, df) {
  predict(fitted_model, df, type="response")
```

```
  }

  # Apply the "my-lasso" apporach to the first stage
  HDCATE.set_first_stage(
    model,
    my_lasso_fit_exp,
    my_lasso_fit_exp,
    my_lasso_fit_ps,
    my_lasso_predict_exp,
    my_lasso_predict_exp,
    my_lasso_predict_ps
  )
```

HDCATE.unset_first_stage

Clear the user-defined first-stage estimating methods

Description

Inverse operation of [HDCATE.set_first_stage](#)

Usage

```
HDCATE.unset_first_stage(model)
```

Arguments

model   an object created via [HDCATE](#)

Value

None.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

# ... manually set user-defined first-stage estimating methods via `HDCATE.set_first_stage`
```

```
# Clear those user-defined methods and use the built-in method
HDCATE.unset_first_stage(model)
```

HDCATE.use_cross_fitting

Use k-fold cross-fitting estimator

Description

Use k-fold cross-fitting estimator

Usage

```
HDCATE.use_cross_fitting(model, k_fold = 5, folds = NULL)
```

Arguments

model	an object created via [HDCATE](#)
k_fold	number of folds
folds	you can manually set the folds, should be a list of index vector

Value

None.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

# for example, use 5-fold cross-fitting estimator
HDCATE.use_cross_fitting(model, k_fold=5)

# alternatively, pass a list of index vector to the third argument to set the folds manually,
# in this case, the second argument k_fold is auto detected, you can pass any value to it.
HDCATE.use_cross_fitting(model, k_fold=2, folds=list(c(1:250), c(251:500)))
```

HDCATE.use_full_sample

Use full-sample estimator

Description

This is the default mode when creating a model via [HDCATE](#)

Usage

```
HDCATE.use_full_sample(model)
```

Arguments

model	an object created via [HDCATE](#)

Value

None.

Examples

```
# get simulation data
n_obs <- 500  # Num of observations
n_var <- 100  # Num of observed variables
n_rel_var <- 4  # Num of relevant variables
data <- HDCATE.get_sim_data(n_obs, n_var, n_rel_var)
# conditional expectation model is misspecified
x_formula <- paste(paste0('X', c(2:n_var)), collapse ='+')
# propensity score model is misspecified
# x_formula <- paste(paste0('X', c(1:(n_var-1))), collapse ='+')

# create a new HDCATE model
model <- HDCATE(data=data, y_name='Y', d_name='D', x_formula=x_formula)

HDCATE.use_full_sample(model)
```

Index