Package ‘CRE’

June 15, 2023

Type Package

Title Interpretable Discovery and Inference of Heterogeneous Treatment Effects

Version 0.2.4

Maintainer Naeem Khoshnevis <nkhoshnevis@g.harvard.edu>


License GPL-3

URL https://github.com/NSAPH-Software/CRE

BugReports https://github.com/NSAPH-Software/CRE/issues

Depends R (>= 3.5.0)

Imports MASS, stats, logger, gbm, randomForest, methods, xgboost, RRF, data.table, xtable, glmnet, bartCause, stabs, stringr, SuperLearner, magrittr, ggplot2, inTrees

Suggests grf, BART, gnm, covr, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Copyright Harvard University

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

NeedsCompilation no
Author  Naeem Khoshnevis [aut, cre] (<https://orcid.org/0000-0003-4315-1426>),
Daniela Maria Garcia [aut] (<https://orcid.org/0000-0003-3226-3561>),
Riccardo Cadei [aut] (<https://orcid.org/0000-0003-2416-8943>),
Kwonsang Lee [aut] (<https://orcid.org/0000-0002-5823-4331>),
Falco Joannes Bargagli Stoffi [aut] (<https://orcid.org/0000-0002-6131-8165>)

Repository  CRAN

Date/Publication  2023-06-14 23:12:08 UTC

R topics documented:

CRE-package .......................................................... 2
cre ................................................................. 3
generate_cre_dataset ............................................. 5
get_logger ......................................................... 7
plot.cre ......................................................... 7
print.cre ....................................................... 8
set_logger ....................................................... 8
summary.cre .................................................... 9

Index  10

CRE-package

The ‘CRE’ package

Description

In health and social sciences, it is critically important to identify subgroups of the study population where a treatment has notable heterogeneity in the causal effects with respect to the average treatment effect. Data-driven discovery of heterogeneous treatment effects (HTE) via decision tree methods has been proposed for this task. Despite its high interpretability, the single-tree discovery of HTE tends to be highly unstable and to find an oversimplified representation of treatment heterogeneity. To accommodate these shortcomings, we propose Causal Rule Ensemble (CRE), a new method to discover heterogeneous subgroups through an ensemble-of-trees approach. CRE has the following features:

1. provides an interpretable representation of the HTE; 2) allows extensive exploration of complex heterogeneity patterns; and 3) guarantees high stability in the discovery. The discovered subgroups are defined in terms of interpretable decision rules, and we develop a general two-stage approach for subgroup-specific conditional causal effects estimation, providing theoretical guarantees.
**cre**

**Author(s)**
Naeem Khoshnevis  
Daniela Maria Garcia  
Riccardo Cadei  
Kwonsang Lee  
Falco Joannes Bargagli Stoffi

**References**

---

**Causal rule ensemble**

**Description**
Performs the Causal Rule Ensemble on a data set with a response variable, a treatment variable, and various features.

**Usage**
cre(y, z, X, method_params = NULL, hyper_params = NULL, ite = NULL)

**Arguments**
- **y**: An observed response vector.
- **z**: A treatment vector.
- **X**: A covariate matrix (or a data frame). Should be provided as numerical values.
- **method_params**: The list of parameters to define the models used, including:
  - **Parameters for Honest Splitting**
    - *ratio_dis*: The ratio of data delegated to rules discovery (default: 0.5).
  - **Parameters for Discovery**
    - *ite_method_dis*: The method to estimate the discovery sample ITE (default: 'aipw').
    - *ps_method_dis*: The estimation model for the propensity score on the discovery subsample (default: 'SL.xgboost').
    - *oreg_method_dis*: The estimation model for the outcome regressions estimate_ite_aipw on the discovery subsample (default: 'SL.xgboost').
  - **Parameters for Inference**
    - *ite_method_inf*: The method to estimate the inference sample ITE (default: 'aipw').
- **ps_method_inf**: The estimation model for the propensity score on the inference subsample (default: 'SL.xgboost').
- **oreg_method_inf**: The estimation model for the outcome regressions in estimate_ite_aipw on the inference subsample (default: 'SL.xgboost').

**hyper_params**

The list of hyper parameters to fine-tune the method, including:

- **intervention_vars**: Intervention-able variables used for rules generation. Use NULL to include all variables (default: NULL).
- **offset**: Name of the covariate to use as offset (i.e. 'x1') for T-Poisson ITE estimation. Use NULL if offset is not used (default: NULL).
- **ntrees_rf**: A number of decision trees for random forest (default: 20).
- **ntrees_gbm**: A number of decision trees for the generalized boosted regression modeling algorithm. (default: 20).
- **node_size**: Minimum size of the trees’ terminal nodes (default: 20).
- **max_nodes**: Maximum number of terminal nodes per tree (default: 5).
- **max_depth**: Maximum rules length (default: 3).
- **replace**: Boolean variable for replacement in bootstrapping for rules generation by random forest (default: TRUE).
- **t_decay**: The decay threshold for rules pruning. Higher values will carry out an aggressive pruning (default: 0.025).
- **t_ext**: The threshold to truncate too generic or too specific (extreme) rules (default: 0.01, range: [0, 0.5]).
- **t_corr**: The threshold to define correlated rules (default: 1, range: (0, +inf)).
- **t_pvalue**: the threshold to define statistically significant rules (default: 0.05, range: (0, 1)).
- **stability_selection**: Whether or not using stability selection for selecting the rules (default: TRUE).
- **cutoff**: Threshold (percentage) defining the minimum cutoff value for the stability scores (default: 0.9).
- **pfer**: Upper bound for the per-family error rate (tolerated amount of falsely selected rules) (default: 1).
- **penalty_rl**: Order of penalty for rules length during LASSO regularization (i.e. 0: no penalty, 1: rules_length, 2: rules_length^2) (default: 1).

**ite**

The estimated ITE vector. If given both the ITE estimation steps in Discovery and Inference are skipped (default: NULL).

**Value**

An S3 object containing:

- A number of Decision Rules extracted at each step (M).
- A data.frame of Conditional Average Treatment Effect decomposition estimates with corresponding uncertainty quantification (CATE).
- A list of method parameters (method_params).
- A list of hyper parameters (hyper_params).
- An Individual Treatment Effect predicted (ite_pred).
generate_cre_dataset

**Note**

- If `intervention_vars` are provided, it’s important to note that the individual treatment effect will still be computed using all covariates.

**Examples**

```r
set.seed(2021)
dataset <- generate_cre_dataset(n = 400, rho = 0, n_rules = 2, p = 10,
                               effect_size = 2, binary_covariates = TRUE,
                               binary_outcome = FALSE, confounding = "no")
y <- dataset["y"]
z <- dataset["z"]
X <- dataset["X"]

method_params <- list(ratio_dis = 0.25,
                       ite_method_dis="aipw",
                       ps_method_dis = "SL.xgboost",
                       oreg_method_dis = "SL.xgboost",
                       ite_method_inf = "aipw",
                       ps_method_inf = "SL.xgboost",
                       oreg_method_inf = "SL.xgboost")

hyper_params <- list(intervention_vars = NULL,
                      offset = NULL,
                      ntrees_rf = 20,
                      ntrees_gbm = 20,
                      node_size = 20,
                      max_nodes = 5,
                      max_depth = 3,
                      t_decay = 0.025,
                      t_ext = 0.025,
                      t_corr = 1,
                      t_pvalue = 0.05,
                      replace = FALSE,
                      stability_selection = TRUE,
                      cutoff = 0.6,
                      pfer = 0.1,
                      penalty_rl = 1)

cre_results <- cre(y, z, X, method_params, hyper_params)
```

---

**generate_cre_dataset**  Generate CRE synthetic data

**Description**

Generates synthetic data with continues or binary outcome.
generate_cre_dataset

generate_cre_dataset(
    n = 1000,
    rho = 0,
    n_rules = 2,
    p = 10,
    effect_size = 2,
    binary_covariates = TRUE,
    binary_outcome = TRUE,
    confounding = "no"
)

Arguments

- **n**: An integer number that represents the number of observations. Non-integer values will be converted into an integer number.
- **rho**: A positive double number that represents the correlation within the covariates (default: 0, range: (0,1)).
- **n_rules**: The number of causal rules. (default: 2, range: 1,2,3,4).
- **p**: The number of covariates (default: 10).
- **effect_size**: The effect size magnitude in (default: 2, range: >=0).
- **binary_covariates**: Whether to use binary or continuous covariates (default: TRUE).
- **binary_outcome**: Whether to use binary or continuous outcomes (default: TRUE).
- **confounding**: Only for continuous outcome, add confounding variables:
  - Linear confounding "lin".
  - Non-linear confounding "nonlin".
  - No confounding "no" (default).

Value

A list of synthetic data containing:
- An outcome vector (y),
- A treatment vector (z),
- A covariates matrix (X) and
- An individual treatment vector (ite)

Note

Set (binary/continuous) covariates domain (binary_covariates). Set (binary/continuous) outcome domain (binary_outcome). Increase complexity in heterogeneity discovery:
- Decreasing the sample size (n),
- adding correlation among variables (rho),
• increasing the number of rules (n_rules),
• increasing the number of covariates (p),
• decreasing the absolute value of the causal effect (effect_size),
• adding linear or not-linear confounders (confounding).

Examples

```r
set.seed(123)
dataset <- generate_cre_dataset(n = 1000, rho = 0, n_rules = 2, p = 10,
  effect_size = 2, binary_covariates = TRUE,
  binary_outcome = TRUE, confounding = "no")
```

---

## get_logger

### Get Logger settings

**Description**

Returns current logger settings.

**Usage**

```r
get_logger()
```

**Value**

Returns a list that includes `logger_file_path` and `logger_level`.

**Examples**

```r
set_logger("mylogger.log", "INFO")
log_meta <- get_logger()
```

---

## plot.cre

### Extend generic plot functions for CRE class

**Description**

A wrapper function to extend generic plot functions for CRE class.

**Usage**

```r
## S3 method for class 'cre'
plot(x, ...)
```
Arguments

x A CRE object.

... Additional arguments passed to customize the plot.

Value

Returns a ggplot2 object, invisibly. This function is called for side effects.

---

print.cre

Extend print function for the CRE object

Description

Prints a brief summary of the CRE object.

Usage

## S3 method for class 'cre'
print(x, verbose = 2, ...)

Arguments

x A cre object from running the CRE function.

verbose Set level of results description details: only results summary 0, results+parameters summary 1, results+parameters+rules summary (default 2).

... Additional arguments passed to customize the results description.

Value

No return value. This function is called for side effects.

---

set_logger

Set Logger settings

Description

Updates logger settings, including log level and location of the file.

Usage

set_logger(logger_file_path = "CRE.log", logger_level = "INFO")
Arguments

logger_file_path
A path (including file name) to log the messages. (Default: CRE.log)

logger_level
The log level. Available levels include:

• TRACE
• DEBUG
• INFO (Default)
• SUCCESS
• WARN
• ERROR
• FATAL

Value
No return value. This function is called for side effects.

Examples

set_logger("Debug")

summary.cre
Print summary of CRE object

Description
Prints a brief summary of the CRE object

Usage

## S3 method for class 'cre'
summary(object, verbose = 2, ...)

Arguments

object
A cre object from running the CRE function.

verbose
Set level of results description details: only results summary 0, results+parameters summary 1, results+parameters+rules summary (default 2).

... Additional arguments passed to customize the results description.

Value
A summary of the CRE object
Index

CRE (CRE-package), 2
cre, 3
CRE-package, 2

generate_cre_dataset, 5
get_logger, 7

plot.cre, 7
print.cre, 8

set_logger, 8
summary.cre, 9