Package 'causalDT'

September 3, 2025

Title Causal Distillation Trees

Version 1.0.0

Description Causal Distillation Tree (CDT) is a novel machine learning method for estimating interpretable subgroups with heterogeneous treatment effects. CDT allows researchers to fit any machine learning model (or metalearner) to estimate heterogeneous treatment effects for each individual, and then ``distills" these predicted heterogeneous treatment effects into interpretable subgroups by fitting an ordinary decision tree to predict the previously-estimated heterogeneous treatment effects. This package provides tools to estimate causal distillation trees (CDT), as detailed in Huang, Tang, and Kenney (2025) <doi:10.48550/arXiv.2502.07275>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo

Imports bcf, dplyr, ggparty, ggplot2, grf, lifecycle, partykit, purrr, R.utils, Rcpp, rlang, rpart, stringr, tibble, tidyselect

URL https://tiffanymtang.github.io/causalDT/

NeedsCompilation yes

Author Tiffany Tang [aut, cre] (ORCID: <https://orcid.org/0000-0002-8079-6867>), Melody Huang [aut], Ana Kenney [aut]

Maintainer Tiffany Tang <ttang4@nd.edu>

Repository CRAN

Date/Publication 2025-09-03 08:00:13 UTC

Contents

`causalDT` *Causal Distillation Trees (CDT)*

Description

This function implements causal distillation trees (CDT), developed in Huang et al. (2025). Briefly, CDT is a two-stage procedure that allows researchers to identify interpretable subgroups with heterogeneous treatment effects. In the first stage, researchers are free to use any machine learning model or metalearner to predict the heterogeneous treatment effects for each individual in the dataset. In the second stage, CDT "distills" these predicted heterogeneous treatment effects into interpretable subgroups by fitting an ordinary decision tree using the predicted heterogeneous treatment effects from the first stage as the response variable.

Usage

```
causalDT(
  X,
  Y,
  Z,
  W = NULL,
  holdout_prop = 0.3,
  holdout_idxs = NULL,
  teacher_model = "causal_forest",
  teacher_predict = NULL,
  student_model = "rpart",
  rpart_control = NULL,
  rpart_prune = c("none", "min", "1se"),
  nfolds_crossfit = NULL,
  nreps_crossfit = NULL,
  B_stability = 100,
  max_depth_stability = NULL,
  ...
)
```

Arguments

X	A tibble, data.frame, or matrix of covariates.
Y	A vector of outcomes.
Z	A vector of treatments.
W	A vector of weights corresponding to treatment propensities.
holdout_prop	Proportion of data to hold out for honest estimation of treatment effects. Used only if holdout_idxs is NULL.
holdout_idxs	A vector of indices to hold out for honest estimation of treatment effects. If NULL, a holdout set of size holdout_prop x nrow(X) is randomly selected.
teacher_model	Teacher model used to estimate individual-level treatment events. Should be either "causal_forest" (default), "bcf", or a function. If "causal_forest", grf::causal_forest() is used as the teacher model. If "bcf", bcf::bcf() is used as the teacher model. Otherwise, the function should take in the named arguments X, Y, Z, optionally W (corresponding to the covariates, outcome, treatment, and propensity weights, respectively), and (optional) additional arguments passed to the function via .... Moreover, the function should return a model object that can be used to predict individual-level treatment effects using teacher_predict(teacher_model, x).
teacher_predict	Function used to predict individual-level treatment effects from the teacher model. Should take in two arguments. as input: the first being the model object returned by teacher_model, and the second being a tibble, data.frame, or matrix of covariates. If NULL, the default is predict().
student_model	Student model used to estimate subgroups of individuals and their corresponding estimated treatment effects. Should be either "rpart" (default) or a function. If "rpart", rpart::rpart() is used. Otherwise, the function should take in two arguments as input: the first being a tibble, data.frame, or matrix of covariates, and the second being a vector of predicted individual-level treatment effects. Moreover, the function should return a list. At a minimum, this list should contain one element named fit that is a model object that can be used to output the leaf membership indices for each observation via predict(student_model, x, type = 'node'). In general, we recommend using the default "rpart".
rpart_control	A list of control parameters for the rpart algorithm. See ? rpart.control for details. Used only if student_model is "rpart".
rpart_prune	Method for pruning the tree. Default is "none". Options are "none", "min", and "1se". If "min", the tree is pruned using the complexity threshold which minimizes the cross-validation error. If "1se", the tree is pruned using the largest complexity threshold which yields a cross-vaidation error within one standard error of the minimum. If "none", the tree is not pruned.
nfolds_crossfit	Number of folds in cross-fitting procedure. If teacher_model is "causal_forest", the default is 1 (no cross-fitting is performed). Otherwise, the default is 2 (one fold for training the teacher model and one fold for estimating the individual-level treatment effects).
nreps_crossfit	Number of repetitions of the cross-fitting procedure. If teacher_model is "causal_forest", the default is 1 (no cross-fitting is performed). Otherwise, the default is 50.

`B_stability` Number of bootstrap samples to use in evaluating stability diagnostics (which can be used to select an appropriate teacher model). Default is 100. Stability diagnostics are only performed if `student_model` is an `rpart` object. If `B_stability` is 0, no stability diagnostics are performed. We refer to Huang et al. (2025) for additional details on using the stability diagnostic to select the teacher model.

`max_depth_stability`
Maximum depth of the decision tree used in evaluating stability diagnostics. If `NULL`, the default is max(4, max depth of fitted student model).

`...` Additional arguments passed to the `teacher_model` function.

Value

A list with the following elements:

`estimate` Estimated subgroup average treatment effects tibble with the following columns:

- leaf_id - Leaf node identifier.
- subgroup - String representation of the subgroup.
- estimate - Estimated conditional average treatment effect for the subgroup.
- variance - Asymptotic variance of the estimated conditional average treatment effect.
- .var1 - Sample variance for treated observations in the subgroup.
- .var0 - Sample variance for control observations in the subgroup.
- .n1 - Number of treated observations in the subgroup.
- .n0 - Number of control observations in the subgroup.
- .sample_idxs - Indices of (holdout) observations in the subgroup.

`student_fit` Output of `student_model()`, which can vary. If `student_model` is "rpart", the output is a list with the following elements:

- fit - The fitted student model. An `rpart` model object.
- tree_info - A data.frame with the tree structure/split information.
- subgroups - A list of subgroups given by their string representation.
- predictions - Student model predictions for the training (non-holdout) data.

`teacher_fit` A list of (cross-fitted) teacher model fits.

`teacher_predictions`
The predicted individual-level treatment effects, averaged across all cross-fitted teacher model.

`teacher_predictions_ls`
A list of predicted individual-level treatment effects from each (cross-fitted) teacher model fit.

`crossfit_idxs_ls`
A list of fold indices used in each cross-fit.

`stability_diagnostics`
A list of stability diagnostics with the following elements:

- jaccard_mean - Vector of mean Jaccard similarity index for each tree depth. The tree depth is given by the vector index.

- jaccard_distribution - List of Jaccard similarity indices across all bootstraps for each tree depth.
- bootstrap_predictions - List of mean student model predictions (for training (non-holdout) data) across all bootstraps for each tree depth.
- bootstrap_predictions_var - List of variance of student model predictions (for training (non-holdout) data) across all bootstraps for each tree depth.
- leaf_ids - List of leaf node identifiers, indicating the leaf membership of each training sample in the (original) fitted student model.

`holdout_idxs` Indices of the holdout set.

References

Huang, M., Tang, T. M., and Kenney, A. M. (2025). Distilling heterogeneous treatment effects: Stable subgroup estimation in causal inference. *arXiv preprint arXiv:2502.07275*.

Examples

```
n <- 50
p <- 3
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z * (X[, 1] > 0) + X[, 2] + rnorm(n, 0.1)

# causal distillation trees using causal forest teacher model

out <- causalDT(X, Y, Z)
```

`estimate_group_cates` *Subgroup CATE estimation.*

Description

This function estimates the conditional average treatment effect for each subgroup given by the fitted decision tree. The conditional average treatment effect is estimated as the difference in the average outcome between treated and control units that fall within each subgroup (i.e., each leaf node in the decision tree).

Usage

```
estimate_group_cates(fit, X, Y, Z)
```

Arguments

fit	Fitted subgroup model used to determine subgroup membership of individuals. Typically, this is a party or rpart object, but any model object that can be used to determine subgroup membership via predict(fit, x, type = 'node') can be used. If predict(fit, x, type = 'node') returns an error, then subgroups are determined based upon the unique values of predict(fit, x).
X	A tibble, data.frame, or matrix of covariates.
Y	A vector of outcomes.
Z	A vector of treatments.

Value

Estimated subgroup average treatment effects tibble with the following columns:

leaf_id	Leaf node identifier.
subgroup	String representation of the subgroup.
estimate	Estimated conditional average treatment effect for the subgroup.
variance	Asymptotic variance of the estimated conditional average treatment effect.
.var1	Sample variance for treated observations in the subgroup.
.var0	Sample variance for control observations in the subgroup.
.n1	Number of treated observations in the subgroup.
.n0	Number of control observations in the subgroup.
.sample_idxs	Indices of (holdout) observations in the subgroup.

Examples

```
n <- 50
p <- 3
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z * (X[, 1] > 0) + X[, 2] + rnorm(n, 0.1)

# causal distillation tree output
out <- causalDT(X, Y, Z)
# compute subgroup CATEs manually
group_cates <- estimate_group_cates(
  out$student_fit$fit,
  X = X[out$holdout_idxs, , drop = FALSE],
  Y = Y[out$holdout_idxs],
  Z = Z[out$holdout_idxs]
)
all.equal(out$estimate, group_cates)
```

`evaluate_subgroup_stability` *Subgroup stability diagnostics*

Description

This function evaluates the stability of the estimated subgroups from causal distillation trees (CDT) using the Jaccard subgroup stability index (SSI), developed in Huang et al. (2025). It is generally recommended to choose teacher models in CDT that result in the most stable subgroups, as indicated by high SSI values.

Usage

```
evaluate_subgroup_stability(
  estimator,
  fit,
  X,
  y,
  Z = NULL,
  rpart_control = NULL,
  B = 100,
  max_depth = NULL
)
```

Arguments

`estimator`	Function used to estimate subgroups of individuals and their corresponding estimated treatment effects. The function should take in `X`, `y`, and optionally `Z` (if input is not `NULL`) and return a model fit (e.g,. output of `rpart`) that can be coerced into a `party` object via `partykit::as_party()`. Typically, `student_rpart` will be used as the `estimator`.
`fit`	Fitted subgroup model (often, the output of `estimator()`). Mainly used to determine an appropriate `max_depth` for the stability diagnostics. If `fit` is not an `rpart` object, stability diagnostics will be skipped.
`X`	A tibble, data.frame, or matrix of covariates.
`y`	A vector of responses to predict.
`Z`	A vector of treatments.
`rpart_control`	A list of control parameters for the `rpart` algorithm. See `? rpart.control` for details.
`B`	Number of bootstrap samples to use in evaluating stability diagnostics. Default is 100.
`max_depth`	Maximum depth of the tree to consider when evaluating stability diagnostics. If `NULL`, the default is max(4, max depth of `fit`).

Value

A list with the following elements:

`jaccard_mean` Vector of mean Jaccard similarity index for each tree depth. The tree depth is given by the vector index.

`jaccard_distribution` List of Jaccard similarity indices across all bootstraps for each tree depth.

`bootstrap_predictions` List of mean student model predictions (for training (non-holdout) data) across all bootstraps for each tree depth.

`bootstrap_predictions_var` List of variance of student model predictions (for training (non-holdout) data) across all bootstraps for each tree depth.

`leaf_ids` List of leaf node identifiers, indicating the leaf membership of each training sample in the (original) fitted student model.

References

Huang, M., Tang, T. M., and Kenney, A. M. (2025). Distilling heterogeneous treatment effects: Stable subgroup estimation in causal inference. *arXiv preprint arXiv:2502.07275*.

Examples

```
n <- 200
p <- 10
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z * (X[, 1] > 0) + X[, 2] + rnorm(n, 0.1)

# run causal distillation trees without stability diagnostics
out <- causalDT(X, Y, Z, B_stability = 0)
# run stability diagnostics
stability_out <- evaluate_subgroup_stability(
  estimator = student_rpart,
  fit = out$student_fit$fit,
  X = X[-out$holdout_idxs, , drop = FALSE],
  y = out$student_fit$predictions
)
```

---

get_rpart_paths *Get decision paths from an rpart model.*

Description

Return the decision paths for each leaf node in an `rpart` model as character strings.

Usage

```
get_rpart_paths(rpart_fit)
```

Arguments

rpart_fit	An rpart object.

Value

A list of character vectors, where each element corresponds to the decision path for a leaf node in the rpart_fit model.

---

get_rpart_tree_info *Get split information from an rpart model.*

Description

Return the split information for each node in an rpart model as a data frame.

Usage

```
get_rpart_tree_info(rpart_fit, X = NULL, digits = getOption("digits"))
```

Arguments

rpart_fit	An rpart object.
X	Optional data frame containing the features used in the rpart model. Only used if the model contains categorical variables.
digits	Number of digits to round the split values to.

Value

A data.frame with information regarding the feature/threshold used for each split in the rpart model.

plot_cdt *Plot causal distillation tree object*

Description

Visualize the subgroups (i.e., the student tree) from a causal distillation tree object.

Usage

```
plot_cdt(cdt, show_digits = 2)
```

Arguments

cdt	A causal distillation tree object, typically the output of causalDT.
show_digits	Number of digits to show in the plot labels. Default is 2.

Value

A plot of the causal distillation tree.

Examples

```
n <- 200
p <- 10
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z * (X[, 1] > 0) + X[, 2] + rnorm(n, 0.1)

cdt <- causalDT(X, Y, Z)
plot_cdt(cdt)
```

plot_jaccard *Plot Jaccard subgroup similarity index (SSI) for causal distillation tree objects*

Description

The Jaccard subgroup similiarity index (SSI) is a measure of the similarity between two candidate partitions of subgroups. To select an appropriate teacher model in CDT, the Jaccard SSI can be used to select the teacher model that recovers the most stable subgroups.

Usage

```
plot_jaccard(...)
```

Arguments

...	Two or more causal distillation tree objects, each is typically the output of causalDT. Arguments should be named (so that they are properly labeled in the resulting plot).

Value

A plot of the Jaccard SSI for each tree depth.

Examples

```
n <- 50
p <- 2
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z * (X[, 1] > 0) + X[, 2] + rnorm(n, 0.1)

# number of bootstraps for stability diagnostics (setting to small value for faster example)
B <- 10

# run CDT with default causal forest teacher model
cdt1 <- causalDT(X, Y, Z, B_stability = B)

# run CDT with custom BCF teacher model
cdt2 <- causalDT(
  X, Y, Z,
  # set BCF training parameters to be small for faster example
  teacher_model = purrr::partial(bcf, nsim = 100, nburn = 10),
  teacher_predict = predict_bcf,
  # set number of cross-fitting replications to be small for faster example
  nreps_crossfit = 5,
  B_stability = B
)
plot_jaccard(`Causal Forest` = cdt1, `BCF` = cdt2)
```

---

rlearner_teacher *Rlearner teacher model wrapper for causal distillation trees*

Description

This is a wrapper function to convert any of the rlearner model functions into a format that can be used as teacher model in the causal distillation tree framework.

Usage

```
rlearner_teacher(rlearner_fun, ...)
```

Arguments

rlearner_fun	One of rlearner::rboost, rlearner::rlasso, or rlearner::rkern to be transformed to teacher model format for CDT.
...	Additional arguments to pass to the base model functions.

Value

Outputs a function that can be used as teacher model in the causal distillation tree framework. The returned function has the signature function(X, Y, Z, W = NULL, ...).

---

student_rpart *Rpart wrapper for causal distillation trees.*

Description

This function is a wrapper around rpart::rpart() that can be easily used as a student model in the causal distillation tree framework.

Usage

```
student_rpart(
  X,
  y,
  method = "anova",
  rpart_control = NULL,
  prune = c("none", "min", "1se"),
  fit_only = FALSE
)
```

Arguments

X	A tibble, data.frame, or matrix of covariates.
y	A vector of responses to predict.
method	Same as method argument in rpart::rpart(). Default is "anova". See rpart::rpart() for more details.
rpart_control	A list of control parameters for the rpart algorithm. See ? rpart.control for details.
prune	Method for pruning the tree. Default is "none". Options are "none", "min", and "1se". If "min", the tree is pruned using the complexity threshold which minimizes the cross-validation error. If "1se", the tree is pruned using the largest complexity threshold which yields a cross-vaidation error within one standard error of the minimum. If "none", the tree is not pruned.
fit_only	Logical. If TRUE, only the fitted model is returned. Default is FALSE.

Value

If `fit_only = TRUE`, the fitted model is returned. Otherwise, a list with the following components is returned:

`fit`	Fitted model. An `rpart` model object.
`tree_info`	Data frame with tree structure/split information.
`subgroups`	List of subgroups given by their string representation.
`predictions`	Student model predictions for the given `X` data.

Index