

# Package ‘causalDT’

May 8, 2026

**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](https://doi.org/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	2
estimate_group_cates	5
evaluate_subgroup_stability	7
get_rpart_paths	8
get_rpart_tree_info	9
plot_cdt	10
plot_jaccard	10
rlearner_teacher	11
student_rpart	12
<b>Index</b>	<b>14</b>

---

causalDT	<i>Causal Distillation Trees (CDT)</i>
----------	--

---

## 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**

<code>X</code>	A tibble, data.frame, or matrix of covariates.
<code>Y</code>	A vector of outcomes.
<code>Z</code>	A vector of treatments.
<code>W</code>	A vector of weights corresponding to treatment propensities.
<code>holdout_prop</code>	Proportion of data to hold out for honest estimation of treatment effects. Used only if <code>holdout_idx</code> is NULL.
<code>holdout_idx</code>	A vector of indices to hold out for honest estimation of treatment effects. If NULL, a holdout set of size <code>holdout_prop</code> x <code>nrow(X)</code> is randomly selected.
<code>teacher_model</code>	Teacher model used to estimate individual-level treatment effects. Should be either "causal_forest" (default), "bcf", or a function. If "causal_forest", <code>grf::causal_forest()</code> is used as the teacher model. If "bcf", <code>bcf::bcf()</code> is used as the teacher model. Otherwise, the function should take in the named arguments <code>X</code> , <code>Y</code> , <code>Z</code> , optionally <code>W</code> (corresponding to the covariates, outcome, treatment, and propensity weights, respectively), and (optional) additional arguments passed to the function via <code>...</code> . Moreover, the function should return a model object that can be used to predict individual-level treatment effects using <code>teacher_predict(teacher_model, x)</code> .
<code>teacher_predict</code>	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 <code>teacher_model</code> , and the second being a tibble, data.frame, or matrix of covariates. If NULL, the default is <code>predict()</code> .
<code>student_model</code>	Student model used to estimate subgroups of individuals and their corresponding estimated treatment effects. Should be either "rpart" (default) or a function. If "rpart", <code>rpart::rpart()</code> 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 <code>fit</code> that is a model object that can be used to output the leaf membership indices for each observation via <code>predict(student_model, x, type = 'node')</code> . In general, we recommend using the default "rpart".
<code>rpart_control</code>	A list of control parameters for the rpart algorithm. See <code>?rpart.control</code> for details. Used only if <code>student_model</code> is "rpart".
<code>rpart_prune</code>	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-validation error within one standard error of the minimum. If "none", the tree is not pruned.
<code>nfolds_crossfit</code>	Number of folds in cross-fitting procedure. If <code>teacher_model</code> 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).
<code>nreps_crossfit</code>	Number of repetitions of the cross-fitting procedure. If <code>teacher_model</code> is "causal_forest", the default is 1 (no cross-fitting is performed). Otherwise, the default is 50.

<code>B_stability</code>	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 <code>student_model</code> is an <code>rpart</code> object. If <code>B_stability</code> 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.
<code>max_depth_stability</code>	Maximum depth of the decision tree used in evaluating stability diagnostics. If <code>NULL</code> , the default is <code>max(4, max depth of fitted student model)</code> .
<code>...</code>	Additional arguments passed to the <code>teacher_model</code> function.

### Value

A list with the following elements:

<code>estimate</code>	Estimated subgroup average treatment effects tibble with the following columns: <ul style="list-style-type: none"> <li><code>leaf_id</code> - Leaf node identifier.</li> <li><code>subgroup</code> - String representation of the subgroup.</li> <li><code>estimate</code> - Estimated conditional average treatment effect for the subgroup.</li> <li><code>variance</code> - Asymptotic variance of the estimated conditional average treatment effect.</li> <li><code>.var1</code> - Sample variance for treated observations in the subgroup.</li> <li><code>.var0</code> - Sample variance for control observations in the subgroup.</li> <li><code>.n1</code> - Number of treated observations in the subgroup.</li> <li><code>.n0</code> - Number of control observations in the subgroup.</li> <li><code>.sample_idx</code> - Indices of (holdout) observations in the subgroup.</li> </ul>
<code>student_fit</code>	Output of <code>student_model()</code> , which can vary. If <code>student_model</code> is "rpart", the output is a list with the following elements: <ul style="list-style-type: none"> <li><code>fit</code> - The fitted student model. An <code>rpart</code> model object.</li> <li><code>tree_info</code> - A <code>data.frame</code> with the tree structure/split information.</li> <li><code>subgroups</code> - A list of subgroups given by their string representation.</li> <li><code>predictions</code> - Student model predictions for the training (non-holdout) data.</li> </ul>
<code>teacher_fit</code>	A list of (cross-fitted) teacher model fits.
<code>teacher_predictions</code>	The predicted individual-level treatment effects, averaged across all cross-fitted teacher model.
<code>teacher_predictions_ls</code>	A list of predicted individual-level treatment effects from each (cross-fitted) teacher model fit.
<code>crossfit_idx</code>	A list of fold indices used in each cross-fit.
<code>stability_diagnostics</code>	A list of stability diagnostics with the following elements: <ul style="list-style-type: none"> <li><code>jaccard_mean</code> - Vector of mean Jaccard similarity index for each tree depth. The tree depth is given by the vector index.</li> </ul>

- `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_idx` 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**

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

**Value**

Estimated subgroup average treatment effects tibble with the following columns:

<code>leaf_id</code>	Leaf node identifier.
<code>subgroup</code>	String representation of the subgroup.
<code>estimate</code>	Estimated conditional average treatment effect for the subgroup.
<code>variance</code>	Asymptotic variance of the estimated conditional average treatment effect.
<code>.var1</code>	Sample variance for treated observations in the subgroup.
<code>.var0</code>	Sample variance for control observations in the subgroup.
<code>.n1</code>	Number of treated observations in the subgroup.
<code>.n0</code>	Number of control observations in the subgroup.
<code>.sample_idx</code>	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$holdout_idx, , drop = FALSE],
  Y = Y[out$holdout_idx],
  Z = Z[out$holdout_idx]
)
all.equal(out$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 <code>X</code> , <code>y</code> , and optionally <code>Z</code> (if input is not <code>NULL</code> ) and return a model fit (e.g., output of <code>rpart</code> ) that can be coerced into a party object via <code>partykit::as_party()</code> . Typically, <code>student_rpart</code> will be used as the estimator.
fit	Fitted subgroup model (often, the output of <code>estimator()</code> ). Mainly used to determine an appropriate <code>max_depth</code> for the stability diagnostics. If <code>fit</code> is not an <code>rpart</code> 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 <code>rpart</code> algorithm. See <code>?rpart.control</code> 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 <code>NULL</code> , the default is <code>max(4, max depth of fit)</code> .

**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$holdout_idx, , drop = FALSE],
  y = out$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	<i>Plot causal distillation tree object</i>
----------	---

---

**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	<i>Plot Jaccard subgroup similarity index (SSI) for causal distillation tree objects</i>
--------------	--

---

**Description**

The Jaccard subgroup similarity 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, ...)`.

---

<code>student_rpart</code>	<i>Rpart wrapper for causal distillation trees.</i>
----------------------------	---

---

**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-validation 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:

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

# Index

`causalDT`, [2](#), [10](#), [11](#)

`estimate_group_cates`, [5](#)

`evaluate_subgroup_stability`, [7](#)

`get_rpart_paths`, [8](#)

`get_rpart_tree_info`, [9](#)

`plot_cdt`, [10](#)

`plot_jaccard`, [10](#)

`rlearner_teacher`, [11](#)

`student_rpart`, [12](#)