Package: causalmlr (via r-universe)

October 14, 2026

Type Package

Title Causal Machine Learning in R

Version 0.1.0

Description Estimation and evaluation of average and conditional average treatment effects (ATE/CATE) from observational data using machine learning. Provides classical estimators (naive difference in means, inverse propensity weighting, doubly robust / AIPW, double machine learning) and meta-learners (S-, T- and X-Learner), all built on top of the 'mlr3' ecosystem so that any regression or classification learner can be plugged in as a nuisance model. Also includes causal model evaluation utilities (ATE error, PEHE, R-Loss) with optional cross-fitting, and a collection of benchmark datasets commonly used for teaching machine learning for causal inference.

License LGPL (>= 2.1)

URL https://github.com/dmachlanski/causalmlr

BugReports https://github.com/dmachlanski/causalmlr/issues

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 4.1.0)

Imports mlr3 (>= 0.14.0), stats

Suggests knitr, lgr, mlr3learners, mlr3tuning, mlr3tuningspaces, ranger, rmarkdown, rpart, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Repository https://dmachlanski.r-universe.dev

Date/Publication 2026-07-16 08:27:15 UTC

RemoteUrl https://github.com/dmachlanski/causalmlr

RemoteRef main

RemoteSha 6a3df30ea501daa48fd4151f7169b5386dfa8f03

Contents

abalone . . . 2
ate . . . 3
ate_dml . . . 4
ate_dr . . . 5
ate_ipw . . . 7
ate_naive . . . 8
cate_ci . . . 9
diabetes . . . 10
dr_learner . . . 11
eps_ate . . . 13
housing . . . 14
ihdp_test . . . 14
ihdp_train . . . 15
jobs_test . . . 15
jobs_train . . . 16
pehe . . . 17
r_learner . . . 18
r_loss . . . 20
rloss_nuisance . . . 21
s_learner . . . 23
sodium . . . 24
spirals . . . 25
synth_test . . . 25
synth_train . . . 26
t_learner . . . 27
x_learner . . . 28

Index **31**

`abalone` *Abalone (regression practice data)*

Description

The classic UCI abalone dataset: predict the number of rings (a proxy for age) of abalone from physical measurements. Included as standard supervised-learning practice data; it has no causal structure.

Usage

```
abalone
```

Format

A data frame with 4,177 rows and 9 columns: `sex` (factor), seven physical measurements and the target `rings` (integer).

Source

https://archive.ics.uci.edu/dataset/1/abalone

Examples

```
data(abalone)
head(abalone)
```

`ate` *Average treatment effect of a fitted CATE model*

Description

Averages the CATE predictions of a fitted meta-learner over the rows of `newdata`:

$$\widehat{ATE} = \frac{1}{n} \sum_i \hat{\tau}(x^{(i)})$$

Usage

```
ate(object, ...)

## S3 method for class 'cate_learner'
ate(object, newdata, ...)
```

Arguments

`object`	A fitted CATE model, e.g. from `s_learner()`.
`...`	Passed on to methods.
`newdata`	A `data.frame` containing at least the covariate columns.

Value

A single numeric value.

Examples

```
library(mlr3)
data(synth_train)
m <- s_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"))
ate(m, synth_train)
```

`ate_dml` *Double machine learning (DML) ATE estimator*

Description

Estimates the average treatment effect in the partially linear model $Y = \theta T + g(X) + \varepsilon$ using the residual-on-residual (orthogonalised) estimator of Chernozhukov et al. (2018). Two nuisance functions are estimated with machine learning and cross-fitting: the conditional outcome mean $m(x) = E[Y|X = x]$ and the propensity score $e(x) = P(T = 1|X = x)$. The ATE is then

$$\hat{\theta} = \frac{\sum_i (T_i - \hat{e}(X_i))(Y_i - \hat{m}(X_i))}{\sum_i (T_i - \hat{e}(X_i))^2}$$

Usage

```
ate_dml(
  data,
  outcome = "y",
  treatment = "t",
  outcome_learner,
  ps_learner,
  covariates = NULL,
  folds = 5,
  ps_trim = NULL
)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`outcome_learner`	An mlr3 regression learner for the conditional outcome mean `m(x) = E[Y | X]` (treatment excluded from the features).
`ps_learner`	An mlr3 classification learner used as the propensity score model, e.g. `mlr3::lrn("classif.log_reg")`. Its predict type is set to `"prob"` automatically.
`covariates`	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
`folds`	Number of cross-fitting folds. Default `5`. Cross-fitting is a core ingredient of DML; `folds = 1` (in-sample nuisance estimates) is allowed but not recommended with flexible learners.
`ps_trim`	Optional trimming for the estimated propensity scores. Either a single value `a` (clips to `[a, 1 - a]`) or a length-2 range. `NULL` (default) applies no trimming.

Details

Note that unlike `ate_dr()`, the outcome model here regresses `Y` on the covariates only (the treatment is excluded), and the model assumes a constant (homogeneous) treatment effect.

Value

An object of class `"causalmlr_ate"`; see `ate_naive()`.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1), C1-C68.

See Also

`ate_dr()`, `ate_ipw()`

Examples

```
library(mlr3)
data(sodium)
set.seed(1)
ate_dml(sodium, outcome = "bp", treatment = "sodium",
        outcome_learner = lrn("regr.rpart"),
        ps_learner = lrn("classif.rpart"),
        ps_trim = 0.01)
```

---

ate_dr *Doubly robust (AIPW) ATE estimator*

Description

Combines an outcome model $\mu(x,t) = E[Y|X=x, T=t]$ with a propensity score model $e(x) = P(T=1|X=x)$ in the augmented inverse propensity weighting (AIPW) estimator:

$$\widehat{ATE} = \frac{1}{n}\sum_i \left[\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{T_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1-T_i)(Y_i - \hat{\mu}_0(X_i))}{1-\hat{e}(X_i)}\right]$$

The estimator is consistent if either the outcome model or the propensity model is correctly specified (hence "doubly robust"). With `folds > 1` both nuisance models are cross-fitted, which yields the cross-fitted AIPW estimator (also known as DML for the interactive/fully heterogeneous model).

Usage

```
ate_dr(
  data,
  outcome = "y",
  treatment = "t",
  outcome_learner,
  ps_learner,
  covariates = NULL,
  folds = 1,
  ps_trim = NULL
)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`outcome_learner`	An mlr3 regression learner for the outcome model, e.g. `mlr3::lrn("regr.ranger")`. The treatment indicator is included as a feature and potential outcomes are predicted by setting it to 0 and 1.
`ps_learner`	An mlr3 classification learner used as the propensity score model, e.g. `mlr3::lrn("classif.log_reg")`. Its predict type is set to `"prob"` automatically.
`covariates`	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
`folds`	Number of cross-fitting folds for the propensity model. `1` (the default) fits and predicts in-sample; values greater than 1 use out-of-fold predictions, which reduces overfitting bias when using flexible learners.
`ps_trim`	Optional trimming for the estimated propensity scores. Either a single value `a` (clips to `[a, 1 - a]`) or a length-2 range. `NULL` (default) applies no trimming.

Value

An object of class `"causalmlr_ate"`; see `ate_naive()`.

References

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427), 846-866.

See Also

`ate_ipw()`, `ate_dml()`

Examples

```
library(mlr3)
data(sodium)
ate_dr(sodium, outcome = "bp", treatment = "sodium",
       outcome_learner = lrn("regr.rpart"),
       ps_learner = lrn("classif.rpart"),
       folds = 5, ps_trim = 0.01)
```

`ate_ipw` *Inverse propensity weighting (IPW) ATE estimator*

Description

Estimates the average treatment effect by weighting outcomes with the inverse of the estimated propensity score $e(x) = P(T = 1|X = x)$:

$$\widehat{ATE} = \frac{1}{n} \sum_i \frac{T_i Y_i}{\hat{e}(X_i)} - \frac{1}{n} \sum_i \frac{(1 - T_i) Y_i}{1 - \hat{e}(X_i)}$$

The propensity score is estimated with any mlr3 classification learner.

Usage

```
ate_ipw(
  data,
  outcome = "y",
  treatment = "t",
  ps_learner,
  covariates = NULL,
  folds = 1,
  ps_trim = NULL
)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`ps_learner`	An mlr3 classification learner used as the propensity score model, e.g. `mlr3::lrn("classif.log_reg")`. Its predict type is set to `"prob"` automatically.
`covariates`	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
`folds`	Number of cross-fitting folds for the propensity model. `1` (the default) fits and predicts in-sample; values greater than 1 use out-of-fold predictions, which reduces overfitting bias when using flexible learners.
`ps_trim`	Optional trimming for the estimated propensity scores. Either a single value `a` (clips to `[a, 1 - a]`) or a length-2 range. `NULL` (default) applies no trimming.

Value

An object of class "causalmlr_ate"; see ate_naive().

See Also

ate_dr(), ate_dml()

Examples

```
library(mlr3)
data(sodium)
ate_ipw(sodium, outcome = "bp", treatment = "sodium",
        ps_learner = lrn("classif.rpart"))
```

---

ate_naive *Naive ATE estimator (difference in means)*

Description

Estimates the average treatment effect as the difference between the mean outcome of the treated and the mean outcome of the controls. This estimator is unbiased only when the treatment is randomised; under confounding it is typically biased and serves as a baseline for the adjusted estimators (ate_ipw(), ate_dr(), ate_dml()).

Usage

```
ate_naive(data, outcome = "y", treatment = "t")
```

Arguments

data	A data.frame with the outcome, treatment and covariates.
outcome	Name of the outcome column. Default "y".
treatment	Name of the binary (0/1) treatment column. Default "t".

Value

An object of class "causalmlr_ate" with elements estimate, se (standard error), n and method. Use coef() to extract the point estimate and confint() for a confidence interval.

See Also

ate_ipw(), ate_dr(), ate_dml()

Examples

```
data(sodium)
ate_naive(sodium, outcome = "bp", treatment = "sodium")
```

`cate_ci` *Bootstrap confidence intervals for predicted CATEs*

Description

Attaches pointwise confidence intervals to the CATE predictions of any fitted meta-learner (`s_learner()`, `t_learner()`, `x_learner()`, `dr_learner()`, `r_learner()`) using the nonparametric bootstrap. For each of `n_boot` replicates the training data is resampled with replacement, the entire learner pipeline (including any nuisance models and cross-fitting) is refitted, and CATEs are predicted for `newdata`. A pointwise interval is then formed from the bootstrap distribution at each row of `newdata`.

Usage

```
cate_ci(
  object,
  newdata,
  train_data,
  n_boot = 200,
  level = 0.95,
  type = c("percentile", "normal"),
  ...
)
```

Arguments

`object`	A fitted meta-learner of class `"cate_learner"`.
`newdata`	A `data.frame` of points at which to predict CATEs and their intervals; must contain the covariate columns.
`train_data`	The `data.frame` the model was fitted on (with the outcome, treatment and covariate columns), used to draw the bootstrap resamples on which the pipeline is refitted.
`n_boot`	Number of bootstrap replicates. Default `200`.
`level`	Confidence level. Default `0.95`.
`type`	Interval type. `"percentile"` (default) uses the empirical quantiles of the bootstrap CATEs; `"normal"` centres the interval on the point estimate and uses a normal-quantile multiple of the bootstrap standard error.
`...`	Ignored.

Details

Because the meta-learners accept arbitrary (and typically non-smooth) machine-learning base learners, no closed-form standard error is available for $\hat{\tau}(x)$; the bootstrap is the one mechanism that applies uniformly across all five learners. Two caveats are worth keeping in mind. First, bootstrap coverage for CATEs estimated with flexible learners is only approximate and can be anti-conservative, so the intervals are best read as a measure of estimation stability rather than exact frequentist coverage. Second, refitting the full pipeline `n_boot` times is computationally expensive. Set the seed with `set.seed()` beforehand for reproducibility.

Value

A `data.frame` with one row per row of `newdata` and columns `estimate` (the point CATE from `object`), `se` (the bootstrap standard error) and `lower`/`upper` (the confidence limits).

See Also

`s_learner()`, `t_learner()`, `x_learner()`, `dr_learner()`, `r_learner()`

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
set.seed(1)
m <- t_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"))
ci <- cate_ci(m, synth_test, train_data = synth_train, n_boot = 50)
head(ci)
```

---

diabetes *Pima Indians diabetes (classification practice data)*

Description

Diagnostic measurements for predicting diabetes onset. Included as standard supervised-learning practice data; it has no causal structure.

Usage

```
diabetes
```

Format

A data frame with 768 rows and 9 columns: eight numeric diagnostic measurements and the target `diabetes` (factor, 0/1).

Examples

```
data(diabetes)
head(diabetes)
```

`dr_learner` *DR-Learner for CATE estimation*

Description

The DR-Learner ("doubly robust learner"; Kennedy, 2023) is a two-stage meta-learner that turns the augmented inverse propensity weighting (AIPW) score used by `ate_dr()` into a model that predicts individual CATEs.

1. Cross-fit the nuisance functions: the propensity score $\hat{e}(x) = P(T = 1|X = x)$ and the potential-outcome models $\hat{\mu}_0(x)$, $\hat{\mu}_1(x)$ from a single outcome model that includes the treatment as a feature.
2. Form the doubly robust pseudo-outcome

$$\psi_i = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{T_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1 - T_i)(Y_i - \hat{\mu}_0(X_i))}{1 - \hat{e}(X_i)}$$

   and regress it on the covariates to obtain the final CATE model $\hat{\tau}(x) = E[\psi|X = x]$.

Usage

```
dr_learner(
  data,
  outcome = "y",
  treatment = "t",
  outcome_learner,
  ps_learner,
  tau_learner = NULL,
  covariates = NULL,
  folds = 5,
  ps_trim = NULL
)

## S3 method for class 'dr_learner'
predict(object, newdata, ...)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.

`outcome_learner`
: An mlr3 regression learner for the stage-1 outcome model, e.g. `mlr3::lrn("regr.ranger")`. The treatment indicator is included as a feature and potential outcomes are predicted by setting it to 0 and 1.

`ps_learner`
: An mlr3 classification learner used as the propensity score model, e.g. `mlr3::lrn("classif.log_reg")`. Its predict type is set to `"prob"` automatically.

`tau_learner`
: Optional mlr3 regression learner for the second-stage pseudo-outcome regression. Defaults to a clone of `outcome_learner`.

`covariates`
: Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.

`folds`
: Number of cross-fitting folds for the nuisance models. Default `5`. Cross-fitting is a core ingredient of the DR-Learner; `folds = 1` (in-sample nuisance estimates) is allowed but not recommended with flexible learners.

`ps_trim`
: Optional trimming for the estimated propensity scores. Either a single value `a` (clips to `[a, 1 - a]`) or a length-2 range. `NULL` (default) applies no trimming.

`object`
: A fitted `dr_learner` model.

`newdata`
: A `data.frame` containing at least the covariate columns.

`...`
: Ignored.

Details

Because the pseudo-outcome has conditional mean equal to the true CATE whenever either nuisance is correctly specified, the second-stage regression targets the CATE directly. Cross-fitting the nuisances (the default `folds = 5`) makes the pseudo-outcome orthogonal to nuisance estimation error, so flexible learners can be used without overfitting bias. Unlike `ate_dr()`, which averages ψ_i to a scalar ATE, the DR-Learner keeps the second-stage model and can therefore predict CATEs on new test data.

Value

A fitted CATE model of class `c("dr_learner", "cate_learner")`. Use `predict()` to obtain CATE estimates for new data and `ate()` for their average.

Methods (by generic)

- `predict(dr_learner)`: Predict CATEs for new data.

References

Kennedy, E. H. (2023). Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2), 3008-3049.

See Also

`s_learner()`, `t_learner()`, `x_learner()`, `r_learner()`, `ate_dr()`

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
set.seed(1)
m <- dr_learner(synth_train, outcome = "y", treatment = "t",
                outcome_learner = lrn("regr.rpart"),
                ps_learner = lrn("classif.rpart"),
                ps_trim = 0.01)
tau_hat <- predict(m, synth_test)
pehe(synth_test$tau, tau_hat)
```

---

eps_ate *Absolute error of an ATE estimate*

Description

Measures the absolute difference between an estimated and the true average treatment effect:

$$\epsilon_{ATE} = |\widehat{ATE} - ATE|$$

Requires the true ATE, so it is only usable with (semi-)synthetic benchmark data such as sodium or ihdp_train.

Usage

```
eps_ate(ate_true, ate_pred)
```

Arguments

ate_true	True ATE (a single number).
ate_pred	Estimated ATE. Either a number or a "causalmlr_ate" object as returned by ate_naive() and friends.

Value

A single non-negative number.

See Also

pehe(), r_loss()

Examples

```
data(sodium)
est <- ate_naive(sodium, outcome = "bp", treatment = "sodium")
eps_ate(1.05, est)
```

housing *Boston housing (regression practice data)*

Description

The Boston housing dataset (Harrison & Rubinfeld, 1978): predict median house value (`MEDV`) from neighbourhood characteristics. Included as standard supervised-learning practice data. Note that this dataset has known ethical concerns (the `B` column encodes a racial statistic) and is provided for teaching purposes only.

Usage

```
housing
```

Format

A data frame with 506 rows and 14 columns.

Examples

```
data(housing)
head(housing)
```

ihdp_test *IHDP semi-synthetic benchmark (test set)*

Description

Held-out portion of the IHDP benchmark; see ihdp_train for details.

Usage

```
ihdp_test
```

Format

A data frame with 75 rows and 28 columns; same columns as ihdp_train.

References

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1), 217-240.

Examples

```
data(ihdp_test)
head(ihdp_test)
```

ihdp_train *IHDP semi-synthetic benchmark (training set)*

Description

The Infant Health and Development Program (IHDP) benchmark introduced by Hill (2011): real covariates describing children and their mothers from a randomised experiment, combined with a simulated outcome, which provides ground-truth treatment effects. The treatment is intensive, high-quality childcare and home visits; the (simulated) outcome mimics future cognitive test scores.

Usage

```
ihdp_train
```

Format

A data frame with 672 rows and 28 columns:

x0-x5 Continuous covariates (standardised).

x6-x24 Binary covariates.

t Binary treatment indicator (0/1).

y Continuous (simulated) outcome.

tau True conditional average treatment effect.

References

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1), 217-240.

Examples

```
data(ihdp_train)
head(ihdp_train)
```

jobs_test *Jobs benchmark (test set)*

Description

Held-out portion of the Jobs benchmark; see jobs_train for details.

Usage

```
jobs_test
```

Format

A data frame with 321 rows and 20 columns; same columns as jobs_train.

References

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, 76(4), 604-620.

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. *Proceedings of the 34th International Conference on Machine Learning*.

Examples

```
data(jobs_test)
head(jobs_test)
```

---

jobs_train *Jobs benchmark (training set)*

Description

The Jobs benchmark (LaLonde, 1986; composition of Shalit et al., 2017) combines a randomised experiment (the National Supported Work programme) with observational controls. The treatment is participation in a job training programme and the outcome is employment status. The e column flags units belonging to the randomised subsample, which enables evaluation on real data via the experimental subset.

Usage

```
jobs_train
```

Format

A data frame with 2,891 rows and 20 columns:

x0-x16 Covariates (standardised continuous and binary).

t Binary treatment indicator: job training (0/1).

y Binary outcome: employment (0/1).

e Indicator of membership in the randomised experimental subsample (0/1).

References

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, 76(4), 604-620.

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. *Proceedings of the 34th International Conference on Machine Learning*.

Examples

```
data(jobs_train)
head(jobs_train)
```

---

pehe *Precision in estimation of heterogeneous effects (PEHE)*

Description

The root mean squared error between true and predicted conditional average treatment effects:

$$PEHE = \sqrt{\frac{1}{n}\sum_i (\tau(x^{(i)}) - \hat{\tau}(x^{(i)}))^2}$$

Requires the true CATEs, so it is only usable with (semi-)synthetic benchmark data such as synth_test or ihdp_test.

Usage

```
pehe(tau_true, tau_pred)
```

Arguments

`tau_true`	Numeric vector of true CATEs.
`tau_pred`	Numeric vector of predicted CATEs (same length).

Value

A single non-negative number.

References

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1), 217-240.

See Also

`eps_ate()`, `r_loss()`

Examples

```
pehe(c(1, 2, 3), c(1.1, 1.8, 3.2))
```

`r_learner` *R-Learner for CATE estimation*

Description

The R-Learner (Nie & Wager, 2021) is the heterogeneous-effect generalisation of the partially linear DML estimator (`ate_dml()`). It builds on the Robinson decomposition of the model $Y = \tau(X)T + g(X) + \varepsilon$:

1. Cross-fit the two nuisance functions, the conditional outcome mean $\hat{m}(x) = E[Y|X = x]$ (the treatment is excluded from the features) and the propensity score $\hat{e}(x) = P(T = 1|X = x)$, and form the residuals $\tilde{Y} = Y - \hat{m}(X)$ and $\tilde{T} = T - \hat{e}(X)$.
2. Estimate the CATE by minimising the R-Loss

$$\hat{\tau} = \arg\min_{\tau} \frac{1}{n} \sum_{i} \left[\tilde{Y}_i - \tilde{T}_i \, \tau(X_i) \right]^2$$

which is fitted as a weighted regression of the pseudo-outcome $\tilde{Y}_i / \tilde{T}_i$ on the covariates with weights $\tilde{T}_i^2$.

Usage

```
r_learner(
  data,
  outcome = "y",
  treatment = "t",
  outcome_learner,
  ps_learner,
  tau_learner = NULL,
  covariates = NULL,
  folds = 5,
  ps_trim = NULL
)

## S3 method for class 'r_learner'
predict(object, newdata, ...)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`outcome_learner`	An mlr3 regression learner for the conditional outcome mean $m(x) = E[Y|X]$ (the treatment is excluded from the features), e.g. `mlr3::lrn("regr.ranger")`.

ps_learner	An mlr3 classification learner used as the propensity score model, e.g. `mlr3::lrn("classif.log_reg")`. Its predict type is set to `"prob"` automatically.
tau_learner	Optional mlr3 regression learner for the second-stage weighted pseudo-outcome regression. Defaults to a clone of `outcome_learner`.
covariates	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
folds	Number of cross-fitting folds for the nuisance models. Default 5. Cross-fitting is a core ingredient of the DR-Learner; `folds = 1` (in-sample nuisance estimates) is allowed but not recommended with flexible learners.
ps_trim	Optional trimming for the estimated propensity scores. Either a single value `a` (clips to `[a, 1 - a]`) or a length-2 range. `NULL` (default) applies no trimming.
object	A fitted `r_learner` model.
newdata	A `data.frame` containing at least the covariate columns.
...	Ignored.

Details

The objective minimised here is exactly the `r_loss()` used elsewhere for model selection, so the R-Learner is the estimator that directly targets that score. When the treatment effect is constant it reduces to `ate_dml()`; unlike that scalar estimator, the R-Learner keeps the second-stage model and can predict CATEs on new test data.

The weighted formulation requires a `tau_learner` that supports observation weights (`"weights" %in% learner$properties`). If it does not, the R-Learner falls back to an unweighted regression of the pseudo-outcome and issues a warning. As with `ate_dml()`, small $\tilde{T}_i$ inflate the pseudo-outcome, so `ps_trim` is recommended with flexible propensity learners.

Value

A fitted CATE model of class `c("r_learner", "cate_learner")`. Use `predict()` to obtain CATE estimates for new data and `ate()` for their average.

Methods (by generic)

- `predict(r_learner)`: Predict CATEs for new data.

References

Nie, X., & Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2), 299-319.

See Also

`s_learner()`, `t_learner()`, `x_learner()`, `dr_learner()`, `ate_dml()`, `r_loss()`

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
set.seed(1)
m <- r_learner(synth_train, outcome = "y", treatment = "t",
               outcome_learner = lrn("regr.rpart"),
               ps_learner = lrn("classif.rpart"),
               ps_trim = 0.01)
tau_hat <- predict(m, synth_test)
pehe(synth_test$tau, tau_hat)
```

`r_loss` *R-Loss: an observable score for CATE models*

Description

Computes the R-Loss (Nie & Wager, 2021), also known as τ-risk$_R$, of a vector of CATE predictions:

$$R\text{-}Loss = \frac{1}{n}\sum_i \left[(Y^{(i)} - \hat{m}(X^{(i)})) - (T^{(i)} - \hat{e}(X^{(i)}))\,\hat{\tau}(X^{(i)})\right]^2$$

Unlike `pehe()`, the R-Loss does not require the true treatment effects, so it can be computed on real observational data. This makes it suitable for hyperparameter tuning and model selection of causal estimators: lower values indicate better CATE models.

Usage

```
r_loss(nuisance, tau_pred)
```

Arguments

`nuisance`	An `"rloss_nuisance"` object created with `rloss_nuisance()`, holding the outcome and treatment residuals.
`tau_pred`	Numeric vector of CATE predictions for the same rows that `nuisance` was computed on.

Value

A single non-negative number (lower is better).

References

Nie, X., & Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2), 299-319.

Machlanski, D., Samothrakis, S., & Clarke, P. (2023). Hyperparameter tuning and model evaluation in causal effect estimation. *arXiv preprint arXiv:2303.01412*.

See Also

rloss_nuisance(), pehe(), eps_ate()

Examples

```
library(mlr3)
data(synth_train)
set.seed(1)
nuis <- rloss_nuisance(synth_train, outcome = "y", treatment = "t",
                       outcome_learner = lrn("regr.rpart"),
                       ps_learner = lrn("classif.rpart"))

# compare two candidate CATE models by their R-Loss
m_s <- s_learner(synth_train, outcome = "y", treatment = "t",
                 learner = lrn("regr.rpart"))
m_t <- t_learner(synth_train, outcome = "y", treatment = "t",
                 learner = lrn("regr.rpart"))
r_loss(nuis, predict(m_s, synth_train))
r_loss(nuis, predict(m_t, synth_train))
```

---

rloss_nuisance *Nuisance models for the R-Loss*

Description

Estimates the two nuisance functions required by the R-Loss (r_loss()): the conditional outcome mean $m(x) = E[Y|X = x]$ and the propensity score $e(x) = P(T = 1|X = x)$, and returns the corresponding residuals $Y - \hat{m}(X)$ and $T - \hat{e}(X)$. By default the nuisance predictions are cross-fitted (out-of-fold), which avoids overfitting bias; set `folds = 1` for simple in-sample estimates.

Usage

```
rloss_nuisance(
  data,
  outcome = "y",
  treatment = "t",
  outcome_learner,
  ps_learner,
  covariates = NULL,
  folds = 5,
  ps_trim = NULL
)
```

Arguments

data	A data.frame with the outcome, treatment and covariates. Typically a held-out validation set that was not used to train the CATE models being evaluated.
outcome	Name of the outcome column. Default "y".
treatment	Name of the binary (0/1) treatment column. Default "t".
outcome_learner	An mlr3 regression learner for m(x) (the treatment is excluded from the features).
ps_learner	An mlr3 classification learner for e(x).
covariates	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
folds	Number of cross-fitting folds. Default 5; 1 disables cross-fitting.
ps_trim	Optional trimming for the estimated propensity scores; see ate_ipw().

Details

Fitting the nuisance models once and reusing the resulting object to score many candidate CATE models (e.g. across a hyperparameter grid) is both faster and methodologically cleaner, since all candidates are compared against the same residuals.

Value

An object of class "rloss_nuisance" with elements y_res, t_res, m_hat and e_hat, to be passed to r_loss().

See Also

r_loss()

Examples

```
library(mlr3)
data(synth_train)
set.seed(1)
nuis <- rloss_nuisance(synth_train, outcome = "y", treatment = "t",
                       outcome_learner = lrn("regr.rpart"),
                       ps_learner = lrn("classif.rpart"))
m <- s_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"))
r_loss(nuis, predict(m, synth_train))
```

`s_learner` *S-Learner for CATE estimation*

Description

The S-Learner ("single learner") fits one outcome model $\mu(x,t) = E[Y|X = x, T = t]$ that includes the treatment indicator as a regular feature. CATEs are obtained by contrasting the predictions with the treatment switched on and off:

$$\hat{\tau}(x) = \hat{\mu}(x,1) - \hat{\mu}(x,0)$$

Usage

```
s_learner(data, outcome = "y", treatment = "t", learner, covariates = NULL)

## S3 method for class 's_learner'
predict(object, newdata, ...)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`learner`	An mlr3 regression learner for the outcome model, e.g. `mlr3::lrn("regr.ranger")`.
`covariates`	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
`object`	A fitted `s_learner` model.
`newdata`	A `data.frame` containing at least the covariate columns.
`...`	Ignored.

Value

A fitted CATE model of class `c("s_learner", "cate_learner")`. Use `predict()` to obtain CATE estimates for new data and `ate()` for their average.

Methods (by generic)

- `predict(s_learner)`: Predict CATEs for new data. Returns a numeric vector $\hat{\tau}(x)$ with one element per row of `newdata` (only the covariate columns are used).

References

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156-4165.

See Also

t_learner(), x_learner(), pehe(), r_loss()

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
m <- s_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"))
tau_hat <- predict(m, synth_test)
pehe(synth_test$tau, tau_hat)
```

---

sodium *Sodium intake and blood pressure (synthetic)*

Description

A simplified simulation of the effect of sodium intake on blood pressure, proposed by Luque-Fernandez et al. (2019). Age confounds both the treatment (high sodium intake) and the outcome (blood pressure). The true ATE used in the data generating process is **1.05**.

Usage

```
sodium
```

Format

A data frame with 10,000 rows and 3 columns:

age Age in years (confounder).

sodium Binary treatment: high sodium intake (1) or not (0).

bp Systolic blood pressure (outcome).

References

Luque-Fernandez, M. A., Schomaker, M., Redondo-Sanchez, D., Jose Sanchez Perez, M., Vaidya, A., & Schnitzer, M. E. (2019). Educational Note: Paradoxical collider effect in the analysis of non-communicable disease epidemiological data. *International Journal of Epidemiology*, 48(2), 640-653.

Examples

```
data(sodium)
ate_naive(sodium, outcome = "bp", treatment = "sodium")
```

spirals *Spirals (classification practice data)*

Description

A synthetic two-dimensional binary classification problem where the two classes form interleaved spirals - a simple example of data that linear models cannot separate. Included as supervised-learning practice data.

Usage

```
spirals
```

Format

A data frame with 2,000 rows and 3 columns: coordinates `x0`, `x1` and the class label `y` (factor, 0/1).

Examples

```
data(spirals)
head(spirals)
```

synth_test *Synthetic CATE benchmark (test set)*

Description

Test covariates and ground-truth treatment effects accompanying synth_train. Contains only the covariates and the true CATE (`tau`), mimicking the deployment setting where a fitted CATE model predicts effects for new individuals.

Usage

```
synth_test
```

Format

A data frame with 250 rows and 6 columns:

x0, x1, x2, x3, x4 Continuous covariates.

tau True conditional average treatment effect.

References

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156-4165.

Examples

```
data(synth_test)
head(synth_test)
```

`synth_train` *Synthetic CATE benchmark (training set)*

Description

Fully synthetic data generated with the data generating process of Künzel et al. (2019), with five Gaussian covariates, a binary treatment and a continuous outcome. The true CATE is a step function of one of the covariates. See synth_test for the accompanying test set with ground truth effects.

Usage

```
synth_train
```

Format

A data frame with 1,000 rows and 7 columns:

x0, x1, x2, x3, x4 Continuous covariates.

t Binary treatment indicator (0/1).

y Continuous outcome.

References

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156-4165.

Examples

```
data(synth_train)
head(synth_train)
```

`t_learner` *T-Learner for CATE estimation*

Description

The T-Learner ("two learners") fits separate outcome models for the control and treated groups, $\mu_0(x) = E[Y|X = x, T = 0]$ and $\mu_1(x) = E[Y|X = x, T = 1]$, and estimates the CATE as their difference:

$$\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$$

Usage

```
t_learner(
  data,
  outcome = "y",
  treatment = "t",
  learner,
  learner1 = NULL,
  covariates = NULL
)

## S3 method for class 't_learner'
predict(object, newdata, ...)
```

Arguments

`data`	A `data.frame` with the outcome, treatment and covariates.
`outcome`	Name of the outcome column. Default `"y"`.
`treatment`	Name of the binary (0/1) treatment column. Default `"t"`.
`learner`	An mlr3 regression learner used for the control-group model (and, if `learner1` is `NULL`, also for the treated-group model).
`learner1`	Optional separate mlr3 regression learner for the treated group. Defaults to a clone of `learner`.
`covariates`	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
`object`	A fitted `t_learner` model.
`newdata`	A `data.frame` containing at least the covariate columns.
`...`	Ignored.

Value

A fitted CATE model of class `c("t_learner", "cate_learner")`; see `s_learner()`.

Methods (by generic)

- `predict(t_learner)`: Predict CATEs for new data.

References

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156-4165.

See Also

`s_learner()`, `x_learner()`

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
m <- t_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"))
tau_hat <- predict(m, synth_test)
pehe(synth_test$tau, tau_hat)
```

`x_learner` *X-Learner for CATE estimation*

Description

The X-Learner (Künzel et al., 2019) proceeds in three stages:

1. Fit group-specific outcome models $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$ as in the T-Learner.
2. Impute individual treatment effects, $D_0 = \hat{\mu}_1(X_0) - Y_0$ for controls and $D_1 = Y_1 - \hat{\mu}_0(X_1)$ for the treated, and regress them on the covariates to obtain $\hat{\tau}_0(x)$ and $\hat{\tau}_1(x)$.
3. Combine the two estimates with propensity score weights:

$$\hat{\tau}(x) = \hat{e}(x)\hat{\tau}_0(x) + (1 - \hat{e}(x))\hat{\tau}_1(x)$$

Usage

```
x_learner(
  data,
  outcome = "y",
  treatment = "t",
  learner,
  ps_learner,
  tau_learner = NULL,
  covariates = NULL
)

## S3 method for class 'x_learner'
predict(object, newdata, ...)
```

Arguments

data	A data.frame with the outcome, treatment and covariates.
outcome	Name of the outcome column. Default "y".
treatment	Name of the binary (0/1) treatment column. Default "t".
learner	An mlr3 regression learner for the stage-1 outcome models.
ps_learner	An mlr3 classification learner for the propensity score model, e.g. mlr3::lrn("classif.log_reg").
tau_learner	Optional mlr3 regression learner for the stage-2 imputed-effect models. Defaults to a clone of learner.
covariates	Optional character vector of covariate columns. Defaults to all columns except the outcome and the treatment.
object	A fitted x_learner model.
newdata	A data.frame containing at least the covariate columns.
...	Ignored.

Details

The X-Learner is particularly effective when the treated and control groups are of very different sizes.

Value

A fitted CATE model of class c("x_learner", "cate_learner"); see s_learner().

Methods (by generic)

- predict(x_learner): Predict CATEs for new data.

References

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156-4165.

See Also

s_learner(), t_learner()

Examples

```
library(mlr3)
data(synth_train)
data(synth_test)
m <- x_learner(synth_train, outcome = "y", treatment = "t",
               learner = lrn("regr.rpart"),
               ps_learner = lrn("classif.rpart"))
tau_hat <- predict(m, synth_test)
pehe(synth_test$tau, tau_hat)
```

Index

∗ **datasets**
abalone, 2
diabetes, 10
housing, 14
ihdp_test, 14
ihdp_train, 15
jobs_test, 15
jobs_train, 16
sodium, 24
spirals, 25
synth_test, 25
synth_train, 26

abalone, 2
ate, 3
ate(), *12*, *19*, *23*
ate_dml, 4
ate_dml(), *6*, *8*, *18, 19*
ate_dr, 5
ate_dr(), *5*, *8*, *11, 12*
ate_ipw, 7
ate_ipw(), *5, 6*, *8*, *22*
ate_naive, 8
ate_naive(), *5, 6*, *8*, *13*

cate_ci, 9
coef(), *8*
confint(), *8*

diabetes, 10
dr_learner, 11
dr_learner(), *9, 10*, *19*

eps_ate, 13
eps_ate(), *17*, *21*

housing, 14

ihdp_test, 14, *17*
ihdp_train, *13, 14*, 15

jobs_test, 15
jobs_train, *15*, 16, *16*

pehe, 17
pehe(), *13*, *20, 21*, *24*
predict(), *12*, *19*, *23*
predict.dr_learner *(*dr_learner*)*, 11
predict.r_learner *(*r_learner*)*, 18
predict.s_learner *(*s_learner*)*, 23
predict.t_learner *(*t_learner*)*, 27
predict.x_learner *(*x_learner*)*, 28

r_learner, 18
r_learner(), *9, 10*, *12*
r_loss, 20
r_loss(), *13*, *17*, *19*, *21, 22*, *24*
rloss_nuisance, 21
rloss_nuisance(), *20, 21*

s_learner, 23
s_learner(), *3*, *9, 10*, *12*, *19*, *27–29*
set.seed(), *10*
sodium, *13*, 24
spirals, 25
synth_test, *17*, 25, *26*
synth_train, *25*, 26

t_learner, 27
t_learner(), *9, 10*, *12*, *19*, *24*, *29*

x_learner, 28
x_learner(), *9, 10*, *12*, *19*, *24*, *28*