Package 'evalHTE'

November 5, 2026

Title Evaluating Heterogeneous Treatment Effects

Version 0.3.0

Description Provides various statistical methods for evaluating heterogeneous treatment effects (HTE) in randomized experiments. The package includes tools to estimate uniform confidence bands for estimation of the group average treatment effect sorted by generic machine learning algorithms (GATES). It also provides the tools to identify a subgroup of individuals who are likely to benefit from a treatment the most ``exceptional responders'' or those who are harmed by it. Detailed reference in Imai and Li (2023) <doi:10.48550/arXiv.2310.07973>.

License MIT + file LICENSE

Depends R (>= 3.5.0), dplyr (>= 1.0.10)

Imports Rcpp, cli, evalITR (>= 1.1.0), ggplot2, ggthemes, rlang, furrr, ggdist, scales, stats, purrr, caret

Encoding UTF-8

Suggests testthat (>= 3.0.0), knitr, rmarkdown, future, grf, magrittr, tibble

VignetteBuilder knitr

LinkingTo Rcpp

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Michael Lingzhi Li [aut, cre], Kosuke Imai [aut], Jialu Li [ctb]

Maintainer Michael Lingzhi Li <mili@hbs.edu>

Repository CRAN

Date/Publication 2026-11-05 01:00:02 UTC

Contents

compute_qoi 2
compute_qoi_user 3

consist.test . . . 3
consistcv.test . . . 4
estimate_hte . . . 5
evaluate_hte . . . 6
GATE . . . 7
GATEcv . . . 8
het.test . . . 9
hetcv.test . . . 10
plot.hte . . . 11
plot_CI . . . 11
plot_CI.hte . . . 12
print.summary.hte . . . 13
print.summary.test_hte . . . 13
summary.hte . . . 14
summary.test_hte . . . 14
test_itr . . . 15
URATE . . . 16

Index **18**

`compute_qoi` *Compute Quantities of Interest (GATE, GATEcv, URATE)*

Description

Compute Quantities of Interest (GATE, GATEcv, URATE)

Usage

```
compute_qoi(fit_obj, algorithms, ...)
```

Arguments

`fit_obj`	An output object of `estimate_itr` function from the `evalITR` package.
`algorithms`	Machine learning algorithms
`...`	Calibration arguments passed to `URATE()`.

compute_qoi_user *Compute Evaluation Metrics from a Prespecified Score Function*

Description

Compute Evaluation Metrics from a Prespecified Score Function

Usage

```
compute_qoi_user(user_hte, Tcv, Ycv, data, ngates, ...)
```

Arguments

user_hte	A function returning scores without fitting on evaluation outcomes.
Tcv	Binary evaluation treatment indicators.
Ycv	Evaluation outcomes.
data	Evaluation data passed to the score function.
ngates	Number of fixed groups.
...	Calibration arguments passed to URATE().

consist.test *The Consistency Test for Grouped Average Treatment Effects (GATEs) in Randomized Experiments*

Description

Tests whether treatment effects are nondecreasing across score groups.

Usage

```
consist.test(D, tau, Y, ngates = 5, nsim = 10000, centered = TRUE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	Continuous score vector.
Y	Outcome vector.
ngates	Number of groups (at least 2).
nsim	Number of Monte Carlo draws (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses consist.test for inference.

Value

A list with test statistic stat and p-value pval.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7)
Y = c(4,5,0,2,4,1,-4,3)
consisttestlist <- consist.test(D,tau,Y,ngates=2)
consisttestlist$stat
consisttestlist$pval
```

---

consistcv.test *The Consistency Test for Grouped Average Treatment Effects (GATEs) under Cross Validation in Randomized Experiments*

Description

Tests whether treatment effects are nondecreasing across cross-validated score groups.

Usage

```
consistcv.test(D, tau, Y, ind, ngates = 5, nsim = 10000, centered = TRUE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	A matrix of scores with one column per fold. Column i contains predictions for all observations from a model trained without fold i.
Y	Outcome vector.
ind	Integer validation-fold labels starting at 1.
ngates	Number of groups (at least 2).
nsim	Number of Monte Carlo draws (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses `consistcv.test` for inference.

Value

A list with test statistic `stat` and p-value `pval`.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = matrix(c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9),nrow = 8, ncol = 2)
Y = c(4,5,0,2,4,1,-4,3)
ind = c(rep(1,4),rep(2,4))
consisttestlist <- consistcv.test(D,tau,Y,ind,ngates=2)
consisttestlist$stat
consisttestlist$pval
```

---

estimate_hte *Evaluate Heterogeneous Treatment Effects*

Description

Evaluate Heterogeneous Treatment Effects

Usage

```
estimate_hte(
  treatment,
  form,
  data,
  algorithms,
  n_folds = 5,
  split_ratio = 0,
  ngates = 5,
  preProcess = NULL,
  weights = NULL,
  trControl = caret::trainControl(method = "none"),
  tuneGrid = NULL,
```

```
  tuneLength = ifelse(trControl$method == "none", 1, 3),
  user_model = NULL,
  SL_library = NULL,
  meta_learner = "slearner",
  ...
)
```

Arguments

treatment	Treatment variable
form	a formula object that takes the form y ~ D + x1 + x2 + ...
data	A data frame that contains the outcome y and the treatment D.
algorithms	List of machine learning algorithms to be used.
n_folds	Number of cross-validation folds. Default is 5.
split_ratio	Split ratio between train and test set under sample splitting. Default is 0.
ngates	The number of groups to separate the data into. The groups are determined by tau. Default is 5.
preProcess	caret parameter
weights	caret parameter
trControl	caret parameter
tuneGrid	caret parameter
tuneLength	caret parameter
user_model	A user-defined function to estimate heterogeneous treatment effects.
SL_library	A list of machine learning algorithms to be used in the super learner.
meta_learner	The type of meta-learner to use (e.g., "slearner", "tlearner"). Default is "slearner".
...	Additional arguments passed to caret::train.

Value

An object of hte class

---

evaluate_hte *Evaluate Heterogeneous Treatment Effects*

Description

Evaluate Heterogeneous Treatment Effects

Usage

```
evaluate_hte(fit, alpha = 0.05, ...)
```

Arguments

fit	Fitted model, usually an output from estimate_hte().
alpha	Miscoverage probability for each uniform band. Default is 0.05.
...	Calibration arguments passed to URATE(), such as L, seed, objective, adjustment, and side. These apply to sample splitting only.

Value

An object of class hte with fixed-group estimates and, for sample splitting, calibrated ranked-subgroup bands. Cross-validation returns fixed-group inference only; uniform bands are unavailable.

GATE *Estimation of the Grouped Average Treatment Effects (GATEs) in Randomized Experiments*

Description

Estimates grouped average treatment effects from a continuous score.

Usage

```
GATE(D, tau, Y, ngates = 5, centered = FALSE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	Continuous score vector.
Y	Outcome vector.
ngates	Number of groups (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses GATE for inference.

Value

A list with group estimates gate and standard errors sd, ordered by increasing score.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7)
Y = c(4,5,0,2,4,1,-4,3)
gatelist <- GATE(D,tau,Y,ngates=2)
gatelist$gate
gatelist$sd
```

GATEcv *Estimation of the Grouped Average Treatment Effects (GATEs) in Randomized Experiments Under Cross Validation*

Description

Estimates grouped average treatment effects from cross-validated scores.

Usage

```
GATEcv(D, tau, Y, ind, ngates = 5, centered = FALSE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	A matrix of scores with one column per fold. Column i contains predictions for all observations from a model trained without fold i.
Y	Outcome vector.
ind	Integer validation-fold labels starting at 1.
ngates	Number of groups (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses GATEcv for inference.

Value

A list with group estimates gate and standard errors sd, ordered by increasing score.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = matrix(c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9),nrow = 8, ncol = 2)
Y = c(4,5,0,2,4,1,-4,3)
ind = c(rep(1,4),rep(2,4))
gatelist <- GATEcv(D, tau, Y, ind, ngates = 2)
gatelist$gate
gatelist$sd
```

---

het.test *The Heterogeneity Test for Grouped Average Treatment Effects (GATEs) in Randomized Experiments*

Description

Tests whether treatment effects are equal across score groups.

Usage

```
het.test(D, tau, Y, ngates = 5, centered = TRUE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	Continuous score vector.
Y	Outcome vector.
ngates	Number of groups (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses het.test for inference.

Value

A list with test statistic stat and p-value pval.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7)
Y = c(4,5,0,2,4,1,-4,3)
hettestlist <- het.test(D,tau,Y,ngates=2)
hettestlist$stat
hettestlist$pval
```

---

hetcv.test *The Heterogeneity Test for Grouped Average Treatment Effects (GATEs) under Cross Validation in Randomized Experiments*

Description

Tests whether treatment effects are equal across cross-validated score groups.

Usage

```
hetcv.test(D, tau, Y, ind, ngates = 5, centered = TRUE)
```

Arguments

D	Binary treatment indicator (0 or 1).
tau	A matrix of scores with one column per fold. Column i contains predictions for all observations from a model trained without fold i.
Y	Outcome vector.
ind	Integer validation-fold labels starting at 1.
ngates	Number of groups (at least 2).
centered	Whether to center outcomes before estimation.

Details

Uses hetcv.test for inference.

Value

A list with test statistic stat and p-value pval.

Author(s)

Michael Lingzhi Li, Technology and Operations Management, Harvard Business School <mili@hbs.edu>, https://www.michaellz.com/;

References

Imai and Li (2022). "Statistical Inference for Heterogeneous Treatment Effects Discovered by Generic Machine Learning in Randomized Experiments",

Examples

```
D = c(1,0,1,0,1,0,1,0)
tau = matrix(c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,-0.5,-0.3,-0.1,0.1,0.3,0.5,0.7,0.9),nrow = 8, ncol = 2)
Y = c(4,5,0,2,4,1,-4,3)
ind = c(rep(1,4),rep(2,4))
hettestlist <- hetcv.test(D,tau,Y,ind,ngates=2)
hettestlist$stat
hettestlist$pval
```

---

plot.hte *Plot the GATE estimate*

Description

Plot the GATE estimate

Usage

```
## S3 method for class 'hte'
plot(x, ...)
```

Arguments

x	An table object. This is typically an output of `evaluate_hte()` function.
...	Further arguments passed to the function.

Value

A plot of ggplot2 object.

---

plot_CI *Plot Confidence Intervals*

Description

A generic function to plot uniform and pointwise confidence intervals for HTE objects.

Usage

```
plot_CI(x, ...)
```

Arguments

x	An object for which a plot is desired.
...	Further arguments passed to methods.

Value

A ggplot2 object displaying uniform and pointwise confidence intervals for heterogeneous treatment effects.

---

plot_CI.hte *Plot Calibrated Uniform and Pointwise Bands*

Description

Plot Calibrated Uniform and Pointwise Bands

Usage

```
## S3 method for class 'hte'
plot_CI(x, alpha = NULL, ...)

## S3 method for class 'urate'
plot_CI(x, alpha = NULL, ...)
```

Arguments

x	An object returned by `evaluate_hte()` or `URATE()`.
alpha	Optional miscoverage probability. `NULL` uses the stored band; a different value recalibrates using the stored reference and seed.
...	Other parameters, currently unused.

Value

A ggplot object. No files are written. Coverage is per model and outcome; it is not simultaneous across plotted panels.

print.summary.hte *Print*

Description

Print

Usage

```
## S3 method for class 'summary.hte'
print(x, ...)
```

Arguments

x	An object of summary.hte class. This is typically an output of summary.hte() function.
...	Other parameters. Currently not supported.

Value

No return value, called for side effects (prints summary tables to console).

print.summary.test_hte *Print*

Description

Print

Usage

```
## S3 method for class 'summary.test_hte'
print(x, ...)
```

Arguments

x	An object of summary.test_hte class. This is typically an output of summary.test_hte() function.
...	Other parameters.

Value

No return value, called for side effects (prints test results to console).

summary.hte *Summarize Fixed Groups and Selected Subgroups*

Description

Summarize Fixed Groups and Selected Subgroups

Usage

```
## S3 method for class 'hte'
summary(object, alpha = NULL, ...)
```

Arguments

object	An object returned by evaluate_hte().
alpha	Optional uniform-band miscoverage probability. NULL uses the stored calibration; another value recalibrates the same reference and seed.
...	Other parameters, currently unused.

Value

A summary.hte list. GATE contains fixed-group estimates with pointwise 95 percent intervals and p-values. URATE contains one selected subgroup per model, its size and fraction, estimate and standard error, uniform bounds, confidence level, and side. Pointwise p-values are omitted for the selected subgroups because selection invalidates their interpretation. URATE is NULL when uniform inference is unavailable under cross-validation.

summary.test_hte *Summarize Heterogeneity and Consistency Tests*

Description

Summarize Heterogeneity and Consistency Tests

Usage

```
## S3 method for class 'test_hte'
summary(object, ...)
```

Arguments

object	An object of test_hte class (typically an output of test_hte() function).
...	Other parameters.

Value

An object of class `summary.test_hte`, which is a list containing:

Consistency A tibble with consistency test results, including columns: algorithm, statistic, and p.value (for sample splitting).

Heterogeneity A tibble with heterogeneity test results, including columns: algorithm, statistic, and p.value (for sample splitting).

Consistency_cv A tibble with consistency test results for cross-validation.

Heterogeneity_cv A tibble with heterogeneity test results for cross-validation.

Note: The output contains either the first two or last two elements depending on whether cross-validation was used.

`test_itr` *Conduct hypothesis tests*

Description

Conduct hypothesis tests

Usage

```
test_itr(model, nsim = 1000, centered = TRUE, ...)
```

Arguments

`model`	Fitted model. Usually an output from `evaluate_hte`.
`nsim`	Number of Monte Carlo simulations used to simulate the null distributions. Default is 1000.
`centered`	Whether to center outcomes before testing.
`...`	Further arguments passed to the function.

Value

An object of `test_itr` class

URATE *Uniform Inference for Ranked Subgroup Effects*

Description

Estimate the average effect among the highest-scoring fraction of a randomized evaluation sample, with a design-corrected uniform band.

Usage

```
URATE(
  D,
  tau,
  Y,
  alpha = 0.05,
  objective = c("cumulative", "gates", "reference"),
  p_min = 0.05,
  L = 10000L,
  seed = 20260921L,
  J = max(2L, floor(length(Y)^(1/3))),
  tie_seed = 101L,
  adjustment = c("none", "difference", "mean_adjusted"),
  side = c("lower", "upper", "two.sided"),
  epsilon = 0.001,
  slope_grid = seq(0, 6, by = 0.025)
)
```

Arguments

D	Binary treatment indicator. Use NULL to estimate ranked means of the observed variable Y, for example a baseline covariate.
tau	Numeric score vector or one-column matrix, with larger scores ranked first. The scoring rule must be prespecified or fitted independently of the evaluation sample.
Y	Finite numeric outcomes. Outcomes are not centered automatically.
alpha	Miscoverage probability, strictly between zero and one half.
objective	Boundary-area objective: "cumulative" (the paper's main specification), "gates" (area over [p_min, 1]), or "reference".
p_min	Lower integration limit for the "gates" objective. This does not restrict the coverage domain or the candidate subgroup sizes.
L	Number of Gaussian calibration draws, at least 100.
seed	Calibration seed. The caller's random-number state is restored.
J	Number of consecutive rank blocks, from 2 through floor(n/2).
tie_seed	Seed for independent random ordering within exact score ties. Strict score comparisons are preserved. The caller's state is restored.

adjustment	"none" estimates the subgroup mean effect; "difference" subtracts the overall mean effect; "mean_adjusted" subtracts the subgroup fraction times the overall mean effect.
side	"lower", "upper", or "two.sided". Two-sided bands are calibrated using absolute Gaussian paths, not two one-sided bounds.
epsilon	Positive minimum ratio of intercept to slope.
slope_grid	Strictly increasing nonnegative slopes starting at zero. An optimum at the upper edge triggers a warning to expand the grid.

Details

Inference conditions on the realized fitted rule and assumes independent sampling and complete randomization with fixed treatment counts. It does not require a consistent CATE model. Inference for an underlying population rule requires the additional score-approximation and margin assumptions in the manuscript. Cross-validated scores are not covered by this construction.

The reference combines blockwise residual variation with a rank bridge and a treatment-count correction. Endpoint adjustments transform each entire path before calibration, including the origin and both limits of each rank interval. The finite slope grid and empirical Gaussian quantiles approximate the asymptotic construction; Monte Carlo and finite-sample error remain. Increase L for final analyses.

Coverage is simultaneous over subgroup fractions for one scoring rule and one outcome. It does not adjust for selecting among models or covariates. For lower or two-sided bands, selected maximizes the lower bound; for upper bands it minimizes the upper bound. Ties select the smallest fraction. A pointwise p-value at that selected size is not valid for post-selection inference. For adjusted curves the endpoint is exactly zero.

Value

An object of class urate. The original fields rate and sd are retained. p gives subgroup fractions; lower and upper are uniform bounds (the unused one-sided bound is infinite); pointwise.lower and pointwise.upper are pointwise references. selected is a rank index, and order maps ranks to input rows. width, beta0, and beta1 describe the optimized boundary. alpha, side, and control record calibration settings. reference and profile support reproducible recalibration.

References

Imai and Li. Statistical Performance Guarantee for Subgroup Identification with Generic Machine Learning. https://arxiv.org/abs/2310.07973.

Examples

```
D <- rep(c(1, 0), 20)
tau <- seq_along(D)
Y <- sin(tau) + D * tau / 20
band <- URATE(D, tau, Y, L = 500)
band$rate[band$selected]
band$lower[band$selected]
```

Index

∗ **evaluation**
consist.test, 3
consistcv.test, 4
GATE, 7
GATEcv, 8
het.test, 9
hetcv.test, 10

compute_qoi, 2
compute_qoi_user, 3
consist.test, 3, *4*
consistcv.test, 4, *5*

estimate_hte, 5
estimate_hte(), *7*
evaluate_hte, 6
evaluate_hte(), *12*, *14*

GATE, *7*, 7
GATEcv, *8*, 8

het.test, *9*, 9
hetcv.test, *10*, 10

plot.hte, 11
plot_CI, 11
plot_CI.hte, 12
plot_CI.urate *(*plot_CI.hte*)*, 12
print.summary.hte, 13
print.summary.test_hte, 13

summary.hte, 14
summary.test_hte, 14

test_itr, 15

URATE, 16
URATE(), *2*, *3*, *7*, *12*