# Package ‘tehtuner’

November 7, 2022

**Title** Fit and Tune Models to Detect Treatment Effect Heterogeneity

**Version** 0.1.1

**Description** Implements methods to fit Virtual Twins models (Foster et al. (2011) <doi:10.1002/sim.4322>) for identifying subgroups with differential effects in the context of clinical trials while controlling the probability of falsely detecting a differential effect when the conditional average treatment effect is uniform across the study population using parameter selection methods proposed in Wolf et al. (2022) <doi:10.1177/17407745221095855>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**Depends** R (>= 3.5.0)

**Imports** party, glmnet, Rdpack, rpart, stringr, SuperLearner, randomForestSRC, earth

**RdMacros** Rdpack

**Suggests** spelling, testthat (>= 3.0.0)

**Language** en-US

**URL** `https://github.com/jackmwolf/tehtuner`

**BugReports** `https://github.com/jackmwolf/tehtuner/issues`

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jack Wolf [aut, cre] (<`https://orcid.org/0000-0002-8919-8740`>)

**Maintainer** Jack Wolf <`jackwolf910@gmail.com`>

**Repository** CRAN

**Date/Publication** 2022-11-07 18:30:02 UTC

# R topics documented:

## get_mnpp *Get the MNPP for the Step 2 model*

### Description

Find the lowest penalty parameter so that the Step 2 model fit for the estimated CATE from Step 1 is constant for all subjects.

### Usage

```
get_mnpp(z, data, step2, Trt, Y)
```

### Arguments

| | |
|---|---|
| `z` | a numeric vector of estimated CATEs from Step 1 |
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `step2` | a character string specifying the Step 2 model. Supports "`lasso`", "`rtree`", or "`ctree`". |

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

## `get_mnpp.ctree` *Get the MNPP for a Conditional Inference Tree*

### Description

Finds the lowest test statistic for a null conditional inference tree

### Usage

```
get_mnpp.ctree(z, data, Trt, Y)
```

### Arguments

`z` a numeric vector of estimated CATEs from Step 1

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

### Value

the MNPP

## `get_mnpp.lasso` *Get the MNPP for a Model fit via Lasso*

### Description

Finds the lowest penalty parameter for a null lasso model.

### Usage

```
get_mnpp.lasso(z, data, Trt, Y)
```

### Arguments

`z` a numeric vector of estimated CATEs from Step 1

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

## get_mnpp.rtree *Get the MNPP for a Regression Tree*

### Description

Finds the lowest complexity parameter for a null regression tree fit

### Usage

```
get_mnpp.rtree(z, data, Trt, Y)
```

### Arguments

| | |
|---|---|
| `z` | a numeric vector of estimated CATEs from Step 1 |
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |

### Value

the MNPP

## get_theta_null *Permute a dataset under the null hypothesis and get the MNPP*

### Description

Permute a dataset under the null hypothesis and get the MNPP

### Usage

```
get_theta_null(data, Trt, Y, zbar, step1, step2, ...)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `zbar` | the estimated marginal treatment effect |
| `step1` | character strings specifying the Step 1 model. Supports either "`lasso`", "`mars`", "`randomforest`", or "`superlearner`". |
| `step2` | a character string specifying the Step 2 model. Supports "`lasso`", "`rtree`", or "`ctree`". |
| `...` | additional arguments to the Step 1 model call. |

### Value

the MNPP for the permuted data set

---

`get_vt1` *Get the appropriate Step 1 estimation function associated with a method*

---

### Description

Get the appropriate Step 1 estimation function associated with a method

### Usage

```
get_vt1(step1)
```

### Arguments

| | |
|---|---|
| `step1` | character strings specifying the Step 1 model. Supports either "`lasso`", "`mars`", "`randomforest`", or "`superlearner`". |

### Value

a function that estimates the CATE through Step 1 of Virtual Twins

---

`get_vt2` *Get the appropriate Step 2 estimation function associated with a method*

---

### Description

Get the appropriate Step 2 estimation function associated with a method

### Usage

```
get_vt2(step2)
```

### Arguments

| | |
|---|---|
| `step2` | a character string specifying the Step 2 model. Supports "`lasso`", "`rtree`", or "`ctree`". |

### Value

a function that fits a model for the CATE through Step 2 of Virtual Twins

## `permute` *Generate a dataset with permuted treatment indicators*

### Description

Sets the marginal treatment effect to zero and then permute all treatment indicators.

### Usage

```
permute(data, Trt, Y, zbar)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `zbar` | the estimated marginal treatment effect |

### Value

a permuted dataset of the same size as `data`

## `tehtuner_example` *Simulated example data*

### Description

Simulated data from a clinical trial with heterogenous treatment effects where the CATE was a function of V1 and V9.

### Usage

```
tehtuner_example
```

### Format

A data frame with 200 rows and 12 columns:

**Trt** Binary treatment indicator

**Y** Continuous response

**V1,V2,V3,V4,V5,V6,V7,V8** Continuous covariates

**V9,V10** Binary covariates

## test_null_theta_ctree *Test if a Value Gives a Null Conditional Inference Tree*

### Description

Fits a conditional inference tree with minimal test statistic `theta` and tests if the tree has more than one terminal node.

### Usage

```
test_null_theta_ctree(theta, z, data, Trt, Y)
```

### Arguments

| | |
|---|---|
| `theta` | a positive double |
| `z` | a numeric vector of estimated CATEs from Step 1 |
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |

### Value

a boolean. `True` if `theta` is large enough to give a null conditional inference tree. `False` otherwise.

## tunevt *Fit a tuned Virtual Twins model*

### Description

`tunevt` fits a Virtual Twins model to estimate factors and subgroups associated with differential treatment effects while controlling the Type I error rate of falsely detecting at least one heterogeneous effect when the treatment effect is uniform across the study population.

### Usage

```
tunevt(
  data,
  Y = "Y",
  Trt = "Trt",
  step1 = "randomforest",
  step2 = "rtree",
  alpha0,
  p_reps,
  keepz = FALSE,
  ...
)
```

## Arguments

data — a data frame containing a response, binary treatment indicators, and covariates.

Y — a string specifying the name of the column of data contains the response.

Trt — a string specifying the name of the column of data contains the treatment indicators.

step1 — character strings specifying the Step 1 model. Supports either "lasso", "mars", "randomforest", or "superlearner".

step2 — a character string specifying the Step 2 model. Supports "lasso", "rtree", or "ctree".

alpha0 — the nominal Type I error rate.

p_reps — the number of permutations to run.

keepz — logical. Should the estimated CATE from Step 1 be returned?

... — additional arguments to the Step 1 model call.

## Details

Virtual Twins is a two-step approach to detecting differential treatment effects. Subjects' conditional average treatment effects (CATEs) are first estimated in Step 1 using a flexible model. Then, a simple and interpretable model is fit in Step 2 to model these estimated CATEs as a function of the covariates.

The Step 2 model is dependent on some tuning parameter. This parameter is selected to control the Type I error rate by permuting the data under the null hypothesis of a constant treatment effect and identifying the minimal null penalty parameter (MNPP), which is the smallest penalty parameter that yields a Step 2 model with no covariate effects. The 1-alpha0 quantile of the distribution of is then used to fit the Step 2 model on the original data.

## Value

an object of class "tunevt".

An object of class "tunevt" is a list containing at least the following components:

call — the matched call

vtmod — the model estimated by the given step2 procedure fit with the permuted tuning parameter for the estimated CATEs from the step1 model. See vt2_lasso, vt2_rtree, or vt2_ctree for specifics.

mnpp — the MNPP for the estimated CATEs from Step 1.

theta_null — a vector of the MNPPs from each permutation under the null hypothesis.

z — if keepz = TRUE, the estimated CATEs from the step1 model.

## References

Foster JC, Taylor JM, Ruberg SJ (2011). "Subgroup identification from randomized clinical trial data." *Statistics in Medicine*, **30**(24), 2867–2880. ISSN 02776715, doi:10.1002/sim.4322, http://doi.wiley.com/10.1002/sim.4322.

Wolf JM, Koopmeiners JS, Vock DM (2022). “A permutation procedure to detect heterogeneous treatment effects in randomized clinical trials while controlling the type I error rate.” *Clinical Trials*, **19**(5), 512-521. ISSN 1740-7745, doi:10.1177/17407745221095855, Publisher: SAGE Publications.

## Examples

```
data(tehtuner_example)
tunevt(
  tehtuner_example, step1 = "lasso", step2 = "rtree",
  alpha0 = 0.2, p_reps = 5
)
```

---

## `tune_theta` *Estimate the penalty parameter for Step 2 of Virtual Twins*

---

### Description

Permutes data under the null hypothesis of a constant treatment effect and calculates the MNPP on each permuted data set. The `1 - alpha` quantile of the distribution is taken.

### Usage

```
tune_theta(data, Trt, Y, zbar, step1, step2, alpha0, p_reps, ...)
```

### Arguments

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

`zbar` the estimated marginal treatment effect

`step1` character strings specifying the Step 1 model. Supports either `"lasso"`, `"mars"`, `"randomforest"`, or `"superlearner"`.

`step2` a character string specifying the Step 2 model. Supports `"lasso"`, `"rtree"`, or `"ctree"`.

`alpha0` the nominal Type I error rate.

`p_reps` the number of permutations to run.

`...` additional arguments to the Step 1 model call.

### Value

the estimated penalty parameter

## `validate_alpha0` *Check if alpha0 is a valid input to tunevt*

### Description

Check if alpha0 is a valid input to tunevt

### Usage

```
validate_alpha0(data, alpha0)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `alpha0` | the nominal Type I error rate. |

### Value

TRUE if `alpha0` is a valid input. Errors otherwise.

## `validate_p_reps` *Check if p_reps is a valid input to tunevt*

### Description

Check if p_reps is a valid input to tunevt

### Usage

```
validate_p_reps(data, p_reps)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `p_reps` | the number of permutations to run. |

### Value

TRUE if `p_reps` is a valid input. Errors otherwise.

## `validate_Trt` *Check if Trt is a valid input to tunevt*

### Description

Check if Trt is a valid input to tunevt

### Usage

```
validate_Trt(data, Trt)
```

### Arguments

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

### Value

TRUE if `Trt` is a valid input. Errors otherwise.

## `validate_Y` *Check if Y is a valid input to tunevt*

### Description

Check if Y is a valid input to tunevt

### Usage

```
validate_Y(data, Y)
```

### Arguments

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Y` a string specifying the name of the column of `data` contains the response.

### Value

TRUE if `Y` is a valid input. Errors otherwise.

## vt1_lasso *Estimate the CATE Using the Lasso for Step 1 of Virtual Twins*

### Description

Estimate the CATE Using the Lasso for Step 1 of Virtual Twins

### Usage

```
vt1_lasso(data, Trt, Y, ...)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `...` | additional arguments to `cv.glmnet` |

### Value

Estimated CATEs for each subject in `data`.

### See Also

Other VT Step 1 functions: `vt1_mars()`, `vt1_rf()`, `vt1_super()`

## vt1_mars *Estimate the CATE Using MARS for Step 1 of Virtual Twins*

### Description

Estimate the CATE Using MARS for Step 1 of Virtual Twins

### Usage

```
vt1_mars(data, Trt, Y, ...)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `...` | additional arguments to `earth` |

## Value

Estimated CATEs for each subject in `data`.

## See Also

Other VT Step 1 functions: `vt1_lasso()`, `vt1_rf()`, `vt1_super()`

---

# `vt1_rf` *Estimate the CATE Using a Random Forest for Step 1 of Virtual Twins*

---

## Description

Estimate the CATE Using a Random Forest for Step 1 of Virtual Twins

## Usage

```
vt1_rf(data, Trt, Y, ...)
```

## Arguments

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

`...` additional arguments to `rfsrc`

## Value

Estimated CATEs for each subject in `data`.

## See Also

Other VT Step 1 functions: `vt1_lasso()`, `vt1_mars()`, `vt1_super()`

## vt1_super *Estimate the CATE Using Super Learner for Step 1 of Virtual Twins*

### Description

Estimate the CATE Using Super Learner for Step 1 of Virtual Twins

### Usage

```
vt1_super(data, Trt, Y, SL.library, ...)
```

### Arguments

| | |
|---|---|
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `SL.library` | Either a character vector of prediction algorithms or a list containing character vector. See `SuperLearner` for more details. |
| `...` | additional arguments to `SuperLearner` |

### Value

Estimated CATEs for each subject in `data`.

### See Also

Other VT Step 1 functions: `vt1_lasso()`, `vt1_mars()`, `vt1_rf()`

## vt2_ctree *Estimate the CATE using a conditional inference tree for Step 2*

### Description

Estimate the CATE using a conditional inference tree for Step 2

### Usage

```
vt2_ctree(z, data, Trt, Y, theta)
```

### Arguments

`z` a numeric vector of estimated CATEs from Step 1

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

`theta` the value of the test statistic that must be exceeded in order to implement a split (`mincriterion`)

### Value

An object of class `BinaryTree-class`. See `BinaryTree-class`.

### See Also

Other VT Step 2 functions: `vt2_lasso()`, `vt2_rtree()`

---

## `vt2_lasso` *Estimate the CATE using the Lasso for Step 2*

---

### Description

Estimate the CATE using the Lasso for Step 2

### Usage

```
vt2_lasso(z, data, Trt, Y, theta)
```

### Arguments

`z` a numeric vector of estimated CATEs from Step 1

`data` a data frame containing a response, binary treatment indicators, and covariates.

`Trt` a string specifying the name of the column of `data` contains the treatment indicators.

`Y` a string specifying the name of the column of `data` contains the response.

`theta` lasso penalty parameter (`lambda`)

### Value

a list of length 3 containing the following elements:

`mod` an object of class `glmnet`. See `glmnet`.

`coefficients` coefficients associated with the penalty parameter `theta`.

`fitted.values` predicted values associated with the penalty parameter `theta`.

### See Also

Other VT Step 2 functions: `vt2_ctree()`, `vt2_rtree()`

## vt2_rtree *Estimate the CATE using a regression tree for Step 2*

### Description

Estimate the CATE using a regression tree for Step 2

### Usage

```
vt2_rtree(z, data, Trt, Y, theta)
```

### Arguments

| | |
|---|---|
| `z` | a numeric vector of estimated CATEs from Step 1 |
| `data` | a data frame containing a response, binary treatment indicators, and covariates. |
| `Trt` | a string specifying the name of the column of `data` contains the treatment indicators. |
| `Y` | a string specifying the name of the column of `data` contains the response. |
| `theta` | tree complexity parameter (`cp`) |

### Value

an object of class `rpart`. See `rpart.object`.

### See Also

Other VT Step 2 functions: `vt2_ctree()`, `vt2_lasso()`

# Index