Package ‘crossurr’

Type Package

Title Cross-Fitting for Doubly Robust Evaluation of High-Dimensional Surrogate Markers

Version 1.0.6

``Doubly robust evaluation of high-dimensional surrogate markers''. Biostatistics <doi:10.1093/biostatistics/kxac020>. You can use these methods to determine how much of the overall treatment effect is explained by a (possibly high-dimensional) set of surrogate markers.

License MIT + file LICENSE

Depends R (>= 3.6.0)

Imports dplyr, gbm, glmnet, glue, parallel, pbapply, purrr, ranger, RCAL, rlang, SIS, stats, SuperLearner, tibble, tidyr

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Denis Agniel [aut, cre], Boris P. Hejblum [aut]

Maintainer Denis Agniel <dagniel@rand.org>

Repository CRAN

Date/Publication 2022-10-21 15:25:07 UTC

R topics documented:

<table>
<thead>
<tr>
<th>sim_data</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>xfr_surrogate</td>
<td>2</td>
</tr>
<tr>
<td>xf_surrogate</td>
<td>4</td>
</tr>
</tbody>
</table>

Index 8
**sim_data**  
A simple function to simulate example data.

**Description**  
A simple function to simulate example data.

**Usage**

```r
sim_data(n, p)
```

**Arguments**

- `n`: number of simulated observations
- `p`: number of simulated variables

**Value**

toy dataset used for demonstrating the methods with outcome `y`, treatment `a`, covariates `x.1`, `x.2`, and surrogates `s.1`, `s.2`, ...

---

**xfr_surrogate**  
A function for estimating the proportion of treatment effect explained using repeated cross-fitting.

**Description**  
A function for estimating the proportion of treatment effect explained using repeated cross-fitting.

**Usage**

```r
xfr_surrogate(
  ds,
  x = NULL,
  s,
  y,
  a,
  splits = 50,
  K = 5,
  outcome_learners = NULL,
  ps_learners = NULL,
  interaction_model = TRUE,
  trim_at = 0.05,
  outcome_family = gaussian(),
  mthd = "superlearner",
)```
n_ptb = 0,
...
}

Arguments

dx a data.frame.
x names of all covariates in dx that should be included to control for confounding
(eg. age, sex, etc). Default is NULL.
s names of surrogates in dx.
y name of the outcome in dx.
a treatment variable name (eg. groups). Expect a binary variable made of 1s and
0s.
splits number of data splits to perform.
K number of folds for cross-fitting. Default is 5.
outcome_learners string vector indicating learners to be used for estimation of the outcome func-
tion (e.g., "SL.ridge"). See the SuperLearner package for details.
ps_learners string vector indicating learners to be used for estimation of the propensity score
function (e.g., "SL.ridge"). See the SuperLearner package for details.
interaction_model logical indicating whether outcome functions for treated and control should be
estimated separately. Default is TRUE.
trim_at threshold at which to trim propensity scores. Default is 0.05.
outcome_family default is 'gaussian' for continuous outcomes. Other choice is 'binomial'
for binary outcomes.
mthd selected regression method. Default is 'superlearner', which uses the SuperLearner
package for estimation. Other choices include 'lasso' (which uses glmnet),
'sis' (which uses SIS), 'cal' (which uses RCAL).
n_ptb Number of perturbations. Default is 0 which means asymptotic standard errors
are used.
... additional parameters (in particular for super_learner)

Value

tibble with columns:

- Rm: estimate of the proportion of treatment effect explained, computed as the median over the
  repeated splits.
- R_se0 standard error for the PTE, accounting for the variability due to splitting.
- R_cil0 lower confidence interval value for the PTE.
- R_cih0 upper confidence interval value for the PTE.
- Dm: estimate of the overall treatment effect, computed as the median over the repeated splits.
xf_surrogate

A function for estimating the proportion of treatment effect explained using cross-fitting.

Description
A function for estimating the proportion of treatment effect explained using cross-fitting.

Usage
xf_surrogate(
  ds,
  x = NULL,
  s,
  y,
)

Examples
n <- 100
p <- 20
q <- 2
wds <- sim_data(n = n, p = p)

if(interactive()){
  lasso_est <- xfr_surrogate(ds = wds,
    x = paste('x.', 1:q, sep = '.'),
    s = paste('s.', 1:p, sep = '.'),
    a = 'a',
    y = 'y',
    splits = 2,
    K = 2,
    trim_at = 0.01,
    mthd = 'lasso',
    ncores = 1)
}
xf_surrogate

5

a,  
K = 5,  
outcome_learners = NULL,  
ps_learners = outcome_learners,  
interaction_model = TRUE,  
trim_at = 0.05,  
outcome_family = gaussian(),  
mthd = "superlearner",  
n_ptb = 0,  
ncores = parallel::detectCores() - 1,  
...

Arguments

data

a data.frame.

x

names of all covariates in ds that should be included to control for confounding  
(eg. age, sex, etc). Default is NULL.

s

names of surrogates in ds.

y

name of the outcome in ds.

a

treatment variable name (eg. groups). Expect a binary variable made of 1s and 0s.

K

number of folds for cross-fitting. Default is 5.

outcome_learners

string vector indicating learners to be used for estimation of the outcome function (e.g., "SL.ridge"). See the SuperLearner package for details.

ps_learners

string vector indicating learners to be used for estimation of the propensity score function (e.g., "SL.ridge"). See the SuperLearner package for details.

interaction_model

logical indicating whether outcome functions for treated and control should be estimated separately. Default is TRUE.

trim_at

threshold at which to trim propensity scores. Default is 0.05.

outcome_family

default is 'gaussian' for continuous outcomes. Other choice is 'binomial' for binary outcomes.

mthd

selected regression method. Default is 'superlearner', which uses the SuperLearner package for estimation. Other choices include 'lasso' (which uses glmnet), 'sis' (which uses SIS), 'cal' (which uses RCAL).

n_ptb

Number of perturbations. Default is 0 which means asymptotic standard errors are used.

ncores

number of cpus used for parallel computations. Default is parallel::detectCores() - 1

... additional parameters (in particular for super_learner)
Value

a tibble with columns:

- `R`: estimate of the proportion of treatment effect explained, equal to `1 - deltahat_s/deltahat`.
- `R_se`: standard error for the PTE.
- `deltahat_s`: residual treatment effect estimate.
- `deltahat_s_se`: standard error for the residual treatment effect.
- `pi_o`: estimate of the proportion of overlap.
- `R_o`: PTE only in the overlap region.
- `R_o_se`: the standard error for `R_o`.
- `deltahat_s_o`: residual treatment effect in overlap region,
- `deltahat_s_se_o`: standard error for `deltahat_s_o`.
- `deltahat`: overall treatment effect estimate.
- `deltahat_se`: standard error for overall treatment effect estimate.
- `delta_diff`: difference between the treatment effects, equal to the numerator of PTE.
- `dd_se`: standard error for `delta_diff`

Examples

```r
n <- 300
p <- 50
q <- 2
wds <- sim_data(n = n, p = p)

if(interactive()){
  sl_est <- xf_surrogate(ds = wds,
                         x = paste('x.', 1:q, sep = ''),
                         s = paste('s.', 1:p, sep = ''),
                         a = 'a',
                         y = 'y',
                         K = 4,
                         trim_at = 0.01,
                         mthd = 'superlearner',
                         outcome_learners = c("SL.mean", "SL.lm", "SL.svm", "SL.ridge"),
                         ps_learners = c("SL.mean", "SL.glm", "SL.svm", "SL.lda"),
                         ncores = 1)

  lasso_est <- xf_surrogate(ds = wds,
                            x = paste('x.', 1:q, sep = ''),
                            s = paste('s.', 1:p, sep = ''),
                            a = 'a',
                            y = 'y',
                            K = 4,
                            trim_at = 0.01,
                            mthd = 'lasso',
                            ncores = 1)
```

xf_surrogate

}

Index

sim_data, 2

xf_surrogate, 4
xfr_surrogate, 2