Package ‘targeted’

Type Package
Title Targeted Inference
Version 0.1.1
Date 2020-05-10
Author Klaus K. Holst [aut, cre]
Maintainer Klaus K. Holst <klaus@holst.it>
Depends R (>= 3.3), lava (>= 1.6.6)
Suggests testthat (>= 0.11), knitr, rmarkdown
URL https://kkholst.github.io/targeted/
BugReports https://github.com/kkholst/targeted/issues
License Apache License (== 2.0)
Imports methods, Rcpp (>= 1.0.0), optimx, futile.logger
LinkingTo Rcpp, RcppArmadillo
LazyLoad yes
NeedsCompilation yes
SystemRequirements C++11
ByteCompile yes
RcppModules riskregmodel, dcmmodel
Encoding UTF-8
RoxygenNote 7.1.0
VignetteBuilder knitr
Repository CRAN
Date/Publication 2020-05-15 15:05:44 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Package</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>targeted-package</td>
<td>2</td>
</tr>
<tr>
<td>ace</td>
<td>2</td>
</tr>
<tr>
<td>riskreg</td>
<td>4</td>
</tr>
<tr>
<td>targeted-class</td>
<td>6</td>
</tr>
<tr>
<td>Us</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ace</td>
<td>AIPW estimator for Average Causal Effect</td>
</tr>
</tbody>
</table>

#### ace

#### Description

Augmented Inverse Probability Weighting estimator for the Average Causal Treatment Effect.

#### Usage

```r
ace(
  formula, 
  data, 
  weights, 
  binary = TRUE, 
  nuisance = NULL, 
  propensity = nuisance, 
  all, 
)```
Ace

```r
missing = FALSE,
labels = NULL,
...
)
```

**Arguments**

- `formula`: Formula (see details below)
- `data`: data.frame
- `weights`: optional frequency weights
- `binary`: Binary response (default TRUE)
- `nuisance`: outcome regression formula
- `propensity`: propensity model formula
- `all`: If TRUE all standard errors are calculated (default TRUE when exposure only has two levels)
- `missing`: If TRUE a missing data (AIPW) estimator is returned
- `labels`: Optional treatment labels
- `...`: Additional arguments to lower level functions

**Details**

The formula may either be specified as: `response ~ treatment | nuisance-formula | propensity-formula`

For example: `ace(y~a | x+z | x*z, data=...)`

Alternatively, as a list: `ace(list(y~a, ~x+z, ~x*z), data=...)`

Or using the nuisance (and propensity argument): `ace(y~a, nuisance=~x+z, ...)`

**Value**

An object of class ‘ace.targeted’ is returned. See `targeted-class` for more details about this class and its generic functions.

**Author(s)**

Klaus K. Holst

**Examples**

```r
m <- lvm(y ~ a+x, a~x)
distribution(m,~ a+y) <- binomial.lvm()
d <- sim(m,1e3,seed=1)
a <- ace(y ~ a, nuisance=~x, data=d)
summary(a)

# Multiple treatments
m <- lvm(y ~ a+x, a~x)
```
distribution(m, ~ y) <- binomial.lvm()
m <- ordinal(m, K=4, ~a)
transform(m, ~a) <- factor
d <- sim(m,1e4)
(a <- ace(y~a|x, data=d))

# Comparison with randomized experiment
m0 <- cancel(m, a~x)
d0 <- sim(m0,2e5)
lm(y~a-1,d0)

# Choosing a different contrast for the association measures
summary(a, contrast=c(2,4))

Risk regression

Description

Risk regression with binary exposure and nuisance model for the odds-product.

Let \( A \) be the binary exposure, \( V \) the set of covariates, and \( Y \) the binary response variable, and define \( p_a(v) = P(Y = 1 \mid A = a, V = v), a \in \{0, 1\} \).

The target parameter is either the relative risk

\[
RR(v) = \frac{p_1(v)}{p_0(v)}
\]

or the risk difference

\[
RD(v) = p_1(v) - p_0(v)
\]

We assume a target parameter model given by either

\[
\log\{RR(v)\} = \alpha'^t v
\]

or

\[
\arctanh\{RD(v)\} = \alpha'^t v
\]

and similarly a working linear nuisance model for the odds-product

\[
\phi(v) = \log\left( \frac{p_0(v)p_1(v)}{(1-p_0(v))(1-p_1(v))} \right) = \beta'^t v
\]

A propensity model for \( E(A = 1 \mid V) \) is also fitted using a logistic regression working model

\[
\text{logit}\{E(A = 1 \mid V = v)\} = \gamma'^t v.
\]

If both the odds-product model and the propensity model are correct the estimator is efficient. Further, the estimator is consistent in the union model, i.e., the estimator is double-robust in the sense that only one of the two models needs to be correctly specified to get a consistent estimate.
Usage

riskreg(
  formula,
  target = NULL,
  nuisance = NULL,
  propensity = nuisance,
  data,
  weights,
  type = "rr",
  optimal = TRUE,
  std.err = TRUE,
  start = NULL,
  semi = TRUE,
  ...
)

Arguments

formula       formula (see details below)
target        (optional) target model (formula)
nuisance      nuisance model (formula)
propensity    propensity model (formula)
data          data.frame
weights       optional weights
type          type of association measure (rd og rr)
optimal       If TRUE optimal weights are calculated
std.err       If TRUE standard errors are calculated
start         optional starting values
semi          Semi-parametric (double-robust) estimate (FALSE gives MLE)
...            additional arguments to unconstrained optimization routine (nlminb)

Details

The 'formula' argument should be given as response ~ exposure | target-formula | nuisance-formula
or response ~ exposure | target | nuisance | propensity

E.g., riskreg(y ~ a | 1 | x+z | x+z,data=...)
Alternatively, the model can be specified using the target, nuisance and propensity arguments: riskreg(y
~ a,target=~1,nuisance=~x+z,...)
The riskreg_fit function can be used with matrix inputs rather than formulas.

Value

An object of class 'riskreg.targeted' is returned. See targeted-class for more details about
this class and its generic functions.
Author(s)
Klaus K. Holst

References

Examples
```r
m <- lvm(a[-2] ~ x,
       lp.target[1] ~ 1,
       lp.nuisance[-1] ~ 2*x)
distribution(m,-a) <- binomial.lvm("logit")
m <- binomial.rr(m, "y","a","lp.target","lp.nuisance")
d <- sim(m,5e2,seed=1)
I <- model.matrix(~1, d)
X <- model.matrix(~1+x, d)
with(d, riskreg_mle(y, a, I, X, type="rr"))

with(d, riskreg_fit(y, a, nuisance=X, propensity=I, type="rr"))
riskreg(y ~ a | 1 | x , data=d, type="rr")

## Model with same design matrix for nuisance and propensity model:
with(d, riskreg_fit(y, a, nuisance=X, type="rr"))
a <- riskreg(y ~ a, nuisance=~x, data=d, type="rr")
a```

description

The functions `riskreg` and `ace` returns an object of the type `targeted`. An object of class `targeted-class` is a list with at least the following components:

- **estimate**: An estimate object with the target parameter estimates (see `estimate.default`)
- **opt**: Object returned from the applied optimization routine
- **npar**: number of parameters of the model (target and nuisance)
- **type**: String describing the model

Value

objects of the S3 class 'targeted'
S3 generics

The following S3 generic functions are available for an object of class `targeted`:

- `coef` Extract target coefficients of the estimated model.
- `vcov` Extract the variance-covariance matrix of the target parameters.
- `iid` Extract the estimated influence function.
- `print` Print estimates of the target parameters.
- `summary` Extract information on both target parameters and estimated nuisance model.

See Also

`riskreg`, `ace`

Examples

```r
## See example(riskreg) for examples
```

Us  For internal use

Description

For internal use

Author(s)

Klaus K. Holst
Index

*Topic package
   targeted-package, 2
*Topic utilities
   Us, 7

ace, 2, 6, 7
ace.targeted(targeted-class), 6
aipw(ace), 2
estimate.default, 6

riskreg, 4, 6, 7
riskreg.targeted(targeted-class), 6
riskreg_fit(riskreg), 4
riskreg_mle(riskreg), 4

targeted(targeted-package), 2
targeted-class, 6
targeted-package, 2

Us, 7