Package 'drcmd'

February 3, 2025

Title Doubly-Robust Causal Inference with Missing Data

Version 0.0.0.9000

Author Keith Barnatchez

Maintainer Keith Barnatchez <keithbarnatchez@gmail.com>

Description Doubly-robust estimation of counterfactual means in the presence of missing data. The drcmd() function estimates counterfactual means for binary point treatments and reports average treatment effects, as well as causal risk ratios and odds ratios for binary outcomes. General missingness patterns in the data are allowed and automatically determined by the function -- the only requirement is that any missingness occurs at random conditional on variables that are always available. For scenarios where non-missingness probabilities are known, as is common in two-phase sampling designs, users can provide the non-missingness probabilities through the Rprobs argument. Users can fit nuisance functions through either highly-adaptive LASSO (HAL) or SuperLearner, the latter of which the user must specify libraries.

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends SuperLearner

R topics documented:

`check_binary` *check_binary*

Description

Check if the outcome variable is 0/1 binary. Return 1 if true, 0 if false

Usage

```
check_binary(x)
```

Arguments

`x`	A numeric vector

Value

A logical value

`check_entry_errors` *Check arguments to drcmd for entry errors*

Description

Checks if the arguments to main function drcmd are correctly specified. Returns TRUE if all checks are passed, throws an error with message otherwise.

Usage

```
check_entry_errors(Y, A, X, W, R, eem_ind, Rprobs, k, nboot)
```

Arguments

`Y`	A vector or data frame containing outcome values
`A`	A vector or data frame containing treatment variable values
`X`	A data frame containing covariate values
`W`	A data frame containing proxy variable values
`R`	A vector containing missingness indicator variable

check_r_ind *check_r_ind*

Description

Check if the missingness indicator R is defined correctly

Usage

```
check_r_ind(data, Y, A, X, W, R)
```

Arguments

data	A data frame
Y	A character string containing outcome variable name
A	A character string containing treatment variable name
X	A character vector containing covariate variable names
W	A character vector containing weight variable names
R	A character string containing randomization variable name

Value

A logical value

clean_crossfit_nuis *Clean nuisance function output from crossfit procedure*

Description

Transforms nuisance output across folds into a dataframe of avgerge prediction at each point for each nuisance function

Usage

```
clean_crossfit_nuis(results)
```

Arguments

results	Nuisance functions output from drcmd_est

Value

A dataframe of averaged nuisance estimates across all folds

clean_learners *Clean SuperLearner libraries*

Description

Internal function for setting SuperLearner libraries before they are passed into estimation procedures. Default libraries from the sl_learners argument are used to fill in missing libraries for any nuisance function with libraries left unspecified

Usage

```
clean_learners(
  default_learners,
  m_learners,
  g_learners,
  r_learners,
  po_learners
)
```

Arguments

sl_learners	Either null, or a character vector containing SuperLearner libraries to use for estimating all nuisance functions. User can alternatively specify libraries for each nuisance function for added flexibility
m_sl_learners	Either null, or a character vector containing SuperLearner libraries to be used for the outcome regression
g_sl_learners	Either null, or a character vector containing SuperLearner libraries to be used for the propensity scores
r_sl_learners	Either null, or a character vector containing SuperLearner libraries to be used for the missingness indicator regression
po_sl_learners	Either null, or a character vector containing SuperLearner libraries to be used for the pseudo outcome regression

Value

A list of

create_folds *Create folds for cross-fitting*

Description

Given length of data, creates k folds and returns a list of all possible train-test pairs

Usage

```
create_folds(n, k)
```

Arguments

n	Length of data
k	Number of desired folds

Value

A list of train-test pairs

---

drcmd *Doubly-robust causal inference with missing data*

Description

Doubly-robust estimation of counterfactual means in the presence of missing and reports average treatment effects, as well as causal risk ratios and odds ratios for binary outcomes. General missingness patterns in the data are allowed and automatically determined by the function – the only requirement is that any missingness occurs at random conditional on variables that are always available. For scenarios where non-missingness probabilities are known, as is common in two-phase sampling designs, users can provide the non-missingness probabilities through the Rprobs argument. Users can fit nuisance functions through either highly-adaptive LASSO (HAL) or SuperLearner, the latter of which the user must specify libraries.

Usage

```
drcmd(
  Y,
  A,
  X,
  W = NA,
  R = NA,
  default_learners = NULL,
  m_learners = NULL,
  g_learners = NULL,
  r_learners = NULL,
  po_learners = NULL,
  eem_ind = FALSE,
  Rprobs = NA,
  k = 1,
  c = 0.01,
  nboot = 0
)
```

Arguments

Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
W	(optional) Dataframe containing variables solely predictive of missingness, but not a cause of the outcome or exposure.

R	(optional) A character string specifying the missingness indicator, where 0 indicates missing data. If not specified, the function will create the missingness indicator by identifying the missingness pattern in the data
m_learners	A character vector containing learners to be used for the outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
g_learners	A character vector containing learners to be used for the propensity score. User can specify 'hal' or a vector of SuperLearner libraries
r_learners	A character vector containing learners to be used for the missingness indicator regression. User can specify 'hal' or a vector of SuperLearner libraries
po_learners	A character vector containing learners to be used for the pseudo-outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
eem_ind	A logical indicating whether to use empirical efficiency maximization
Rprobs	A vector of probabilities for the missingness indicator. Only suitable for study designs where researcher controls mechanism by which variables are missing (e.g. two-phase sample designs). Defaults to NA, in which case missingness probabilities are estimated.
k	A numeric indicating the number of folds for cross-fitting
c	Cutoff for treatment and complete case propensity scores. Estimates outside of c, 1-c are set to c or 1-c, respectively
nboot	A numeric indicating the number of desired bootstrap samples. If >0, uses bootstrap to obtain SEs. If =0, uses asymptotic analytical SEs.
deafult_learners	A character vector containing SuperLearner libraries to use for estimating all nuisance functions. User can alternatively specify libraries for each nuisance function for added flexibility

Value

An S3 object of class drcmd containing estimation results, information on the missing data structure, and parameters used in the estimation

Examples

```
n <- 1e3
X <- rnorm(n) ; A <- rbinom(n,1,plogis(X)) ; Y <- rnorm(n) + A + X
Ystar <- Y + rnorm(n)/2 ; R <- rbinom(n,1,plogis(X)) ; X <- as.data.frame(X)
Y[R==0] <- NA

results <- drcmd(Y,A,X,R=R)
```

---

drcmd_est *Obtain doubly-robust counterfactual mean estimates through cross-fitting*

Description

Outer function for estimating counterfactual means through cross-fitting. Calls drcmd_est_fold to obtain estimates for each cross-fitting fold

Usage

```
drcmd_est(
  Y,
  A,
  X,
  Z,
  R,
  m_learners,
  g_learners,
  r_learners,
  po_learners,
  eem_ind,
  Rprobs,
  k,
  c
)
```

Arguments

Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
Z	Dataframe containing all variables that are never subject to missingness
R	Binary missingness indicator
m_learners	A character vector containing learners to be used for the outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
g_learners	A character vector containing learners to be used for the propensity score. User can specify 'hal' or a vector of SuperLearner libraries
r_learners	A character vector containing learners to be used for the missingness indicator regression. User can specify 'hal' or a vector of SuperLearner libraries
po_learners	A character vector containing learners to be used for the pseudo-outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
eem_ind	A logical indicating whether to use empirical efficiency maximization
Rprobs	A vector of probabilities for R
k	A numeric indicating the number of folds for cross-fitting
data	A data frame

Value

A list of point estimates and standard errors for all estimands considered

drcmd_est_fold *Calculate point estimates within a single cross-fitting fold*

Description

Outer function for obtaining point estimates for single fold

Usage

```
drcmd_est_fold(
  splits,
  Y,
  A,
  X,
  Z,
  R,
  m_learners,
  g_learners,
  r_learners,
  po_learners,
  eem_ind,
  Rprobs,
  c
)
```

Arguments

splits	A list of train/test indices
Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
Z	Dataframe containing all variables that are never subject to missingness
R	Binary missingness indicator
m_learners	A character vector containing learners to be used for the outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
g_learners	A character vector containing learners to be used for the propensity score. User can specify 'hal' or a vector of SuperLearner libraries
r_learners	A character vector containing learners to be used for the missingness indicator regression. User can specify 'hal' or a vector of SuperLearner libraries
po_learners	A character vector containing learners to be used for the pseudo-outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
eem_ind	A logical indicating whether to use empirical efficiency maximization
Rprobs	A vector of probabilities for R
c	

Value

A list of results from estimation on current fold

est_g *Estimate the propensity score*

Description

Function for obtaining propensity score estimates

Usage

```
est_g(idx, A, X, R, kappa_hat, g_learners)
```

Arguments

idx	Indices to carry out estimation over
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
R	Binary missingness indicator, where 0 indicates missing data
g_learners	A character vector containing the names of the learners for estimation
Y	Outcome variable. Can be continuous or binary

Value

A list containing the estimate of EY|A=a,X,W

Examples

```
## Not run:
n <- 1000
X <- rnorm(n)
A <- rbinom(n,1,plogis(X))
R <- rbinom(n,1,plogis(X))
X <- data.frame(X)
g_learners <- c('SL.glm','SL.gam')
est_g(idx=1:n, A=A, X=X, R=R, kappa_hat=kappa_hat, g_learners=g_learners)

## End(Not run)
```

est_kappa *Estimate complete case propensity scores*

Description

Function for obtaining complete case propensity score estimates

Usage

```
est_kappa(idx, Z, R, r_learners)
```

Arguments

idx	Indices to carry out estimation over
Z	Dataframe containing all non-missing variables
R	Binary missingness indicator, where 0 indicates missing data
r_learners	A character vector specifying learners to be used for estimation

Value

A numeric vector containing the estimate of ER|Z

Examples

```
## Not run:
n <- 1000
Z <- rnorm(n)
R <- rbinom(n,1,plogis(Z))
Z <- data.frame(Z)
g_learners <- c('SL.glm','SL.gam')
est_g(idx=1:n, Z=Z, R=R, r_learners=r_learners)

## End(Not run)
```

---

est_m_a *Function for estimating outcome regression*

Description

Function for obtaining estimate of EY|A=a,X

Usage

```
est_m_a(idx, Y, A, X, R, kappa_hat, m_learners)
```

Arguments

idx	Indices to carry out estimation over
Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
R	Binary missingness indicator, where 0 indicates missing data
kappa_hat	A numeric vector containing the fitted values of kappa
m_learners	A character vector containing the names of the superlearner algorithms

Value

A list containing the estimate of EY|A=a,X

est_psi *Obtain estimates for current cross fitting fold*

Description

Function for obtaining counterfactual mean estimates for the current fold of the cross-fitting procedure

Usage

```
est_psi(idx, R, Z, kappa_hat, phi_hat, varphi_hat)
```

Arguments

idx	A data frame
R	A character string containing randomization variable name
Z	A character vector containing the names of the variables in Z
kappa_hat	A numeric vector containing the fitted values of kappa
phi_hat	A list containing the estimate of Ephi|Z
varphi_hat	A list containing the estimate of Ephi|Z

Value

A list of point estimates and standard errors for counterfactual means and various counterfactual contrasts

est_varphi *Perform pseudo-outcome regression with conventional loss function*

Description

Outer function for obtaining estimate of Ephi|Z

Usage

```
est_varphi(idx, R, Z, phi_1_hat, phi_0_hat, po_learners)
```

Arguments

idx	Indices to carry out estimation over
R	Binary missingness indicator, where 0 indicates missing data
Z	Dataframe containing all non-missing variables
phi_1_hat	A numeric vector containing the fitted values of phi under A=1
phi_0_hat	A numeric vector containing the fitted values of phi under A=0
po_learners	A character vector containing the names of the superlearner algorithms

Value

A list containing the estimate of Ephi|Z for a=0 and a=1

est_varphi_eem *Perform pseudo-outcome regression with empirical efficiency maximization*

Description

Function for obtaining estimate of Ephi_a|Z via empirical efficiency maximization

Usage

```
est_varphi_eem(idx, R, Z, phi_1_hat, phi_0_hat, kappa_hat, po_learners)
```

Arguments

R	Binary missingness indicator, where 0 indicates missing data
Z	Dataframe containing all non-missing variables
phi_1_hat	Vector of predicted phi under A=1
phi_0_hat	Vector of predicted phi under A=0
po_learners	A character vector containing the names of the superlearner algorithms
data	A data frame
eem_ind	A logical value indicating whether to estimate via EEM

Value

A list containing the estimate of Ephi_a|Z for a=0 and a=1

est_varphi_main *Perform pseudo-outcome regression*

Description

Function for obtaining estimate of Ephi_a|Z through pseudo-outcome regression. Calls inner function to perform estimation, depending on whether user wishes to perform empirical efficiency maximization or not

Usage

```
est_varphi_main(
  idx,
  R,
  Z,
  phi_1_hat,
  phi_0_hat,
  kappa_hat,
  eem_ind,
  po_learners
)
```

Arguments

idx	Indices to carry out estimation over
R	Binary missingness indicator, where 0 indicates missing data
Z	Dataframe containing all non-missing variables
eem_ind	A logical value indicating whether to estimate via EEM
Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
phi_hat	A list containing the estimate of phi

Value

A list containing the estimate of Ephi_alZ for a=0 and a=1

---

find_missing_pattern *find_missing_pattern*

Description

Find the missing pattern in the data

Usage

```
find_missing_pattern(Y, A, X, W)
```

Arguments

Y	A vector or data frame containing outcome values
A	A vector or data frame containing treatment variable values
X	A data frame containing covariate values
W	A data frame containing proxy variable values
data	A data frame
R	A vector containing missingness indicator variable

Value

A character string containing the missing pattern

`get_nuisance_ests` *Obtain nuisance function estimates*

Description

Function for obtaining estimates of all relevant nuisance functions

Usage

```
get_nuisance_ests(
  idx,
  Y,
  A,
  X,
  Z,
  R,
  m_learners,
  g_learners,
  r_learners,
  Rprobs,
  c
)
```

Arguments

`idx`	Indices to carry out estimation over
`Y`	Outcome variable. Can be continuous or binary
`A`	A binary treatment variable (1=treated, 0=control)
`X`	Dataframe containing baseline covariates
`Z`	Dataframe containing all non-missing variables
`R`	Binary missingness indicator, where 0 indicates missing data
`m_learners`	A character vector containing learners to be used for the outcome regression. User can specify 'hal' or a vector of SuperLearner libraries
`g_learners`	A character vector containing learners to be used for the propensity score. User can specify 'hal' or a vector of SuperLearner libraries
`r_learners`	A character vector containing learners to be used for the missingness indicator regression. User can specify 'hal' or a vector of SuperLearner libraries
`Rprobs`	A vector of probabilities for R
`eem_ind`	A logical indicating whether to use empirical efficiency maximization

Value

A list of nuisance estimates

get_phi_hat *Contruct full data EIF estimate*

Description

Function for constructing estimate of the full data EIF, given propensity score and outcome model fits

Usage

```
get_phi_hat(Y, A, X, R, g_hat, m_a_hat, kappa_hat)
```

Arguments

Y	Outcome variable. Can be continuous or binary
A	A binary treatment variable (1=treated, 0=control)
X	Dataframe containing baseline covariates
R	Binary missingness indicator
g_hat	Propensity score predictions
m_a_hat	List of outcome predictions under A=0/1
kappa_hat	Missingness probabilities
Z	Dataframe containing all variables that are never subject to missingness

plot.drcmd *Plot results from drcmd object*

Description

S3 method for plotting results from drcmd object. Plots are available for the following: (1) psuedo outcome regression fit, (2) influence curve distribution, (3) treatment propensity score distribution, and (4) complete-case propensity score distribution. When type='All' (the default), user can view all four plots in succession interactively

Usage

```
## S3 method for class 'drcmd'
plot(x, type = "All")
```

Arguments

x	An object of class drcmd
type	Character denoting type of plot to generate. Must be one of 'All', 'PO', 'IC', 'g_hat', 'r_hat'

Value

No return value. Called for plotting results from drcmd object

print.drcmd *Print drcmd object*

Description

S3 method for printing drcmd objects. Provides concise summary of results, and prints values of optional arguments. Use summary() function for more detailed summary of results.

Usage

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

Arguments

x	An object of class drcmd

summary.drcmd *Summarize results from drcmd*

Description

S3 method for summarizng drcmd results. Provides detailed summary of estimation output, while also providing information on missingness mechanism. Can optionally print out values of user-supplied arguments by setting detail=TRUE

Usage

```
## S3 method for class 'drcmd'
summary(x, detail = FALSE, ...)
```

Arguments

detail	Logical. If TRUE, print out values of user-supplied arguments
results	An object of class drcmd

Value

No return value. Called for printing a detailed results summary

truncate_g *Truncate treatment propensity scores*

Description

Truncate propensity scores to interval c, 1-c

Usage

```
truncate_g(x, c = 0.01)
```

Arguments

x	A vector of treatment propensity scores

Value

A vector of treatment propensity scores truncated to interval c, 1-c

truncate_r *Truncate complete case propensity scores*

Description

Truncate propensity scores to interval c, 1-c

Usage

```
truncate_r(x, c = 0.01)
```

Arguments

x	A vector of complete case propensity scores

Value

A vector of complete propensity scores truncated to interval c, 1-c

Index