

Package ‘CIMTx’

November 30, 2021

Type Package

Title Causal Inference for Multiple Treatments with a Binary Outcome

Version 1.1.0

Description

Different methods to conduct causal inference for multiple treatments with a binary outcome, including regression adjustment, vector matching, Bayesian additive regression trees, targeted maximum likelihood and inverse probability of treatment weighting using different generalized propensity score models such as multinomial logistic regression, generalized boosted models and super learner. For more details, see the paper by Hu et al. <[doi:10.1177/0962280220921909](https://doi.org/10.1177/0962280220921909)>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports nnet (>= 7.3-16), BART (>= 2.9), twang (>= 2.5), arm (>= 1.2-12), dplyr (>= 1.0.7), Matching (>= 4.9-11), magrittr (>= 2.0.1), WeightIt (>= 0.12.0), tmle (>= 1.5.0.2), tidyr (>= 1.1.4), stats, ggplot2 (>= 3.3.5), cowplot (>= 1.1.1), mgcv (>= 1.8-38), metR (>= 0.11.0), stringr (>= 1.4.0), SuperLearner (>= 2.0-28), foreach (>= 1.5.1), doParallel (>= 1.0.16)

NeedsCompilation no

Author Liangyuan Hu [aut],
Chenyang Gu [aut],
Michael Lopez [aut],
Jiayi Ji [aut, cre]

Maintainer Jiayi Ji <Jiayi.Ji@mountsinai.org>

Repository CRAN

Date/Publication 2021-11-29 23:10:09 UTC

R topics documented:

ce_estimate	2
data_sim	4
plot.CIMTx_IPTW	6

plot.CIMTx_sa_grid	7
sa	8
summary.CIMTx_ATE_posterior	10
summary.CIMTx_ATE_sa	11
summary.CIMTx_ATT_posterior	13
summary.CIMTx_ATT_sa	14

Index	16
--------------	-----------

ce_estimate	<i>Causal inference with multiple treatments using observational data</i>
-------------	---

Description

The function `ce_estimate` implements the 6 different methods for causal inference with multiple treatments using observational data.

Usage

```
ce_estimate(
  y,
  x,
  w,
  method,
  formula = NULL,
  discard = FALSE,
  estimand,
  trim_perc = NULL,
  SL.library,
  reference_trt,
  boot = FALSE,
  nboots,
  verbose_boot = TRUE,
  ndpost = 1000,
  caliper = 0.25,
  n_cluster = 5,
  ...
)
```

Arguments

<code>y</code>	A numeric vector (0, 1) representing a binary outcome.
<code>x</code>	A dataframe, including all the covariates but not treatments.
<code>w</code>	A numeric vector representing the treatment groups.
<code>method</code>	A character string. Users can selected from the following methods including "RA", "VM", "BART", "TMLE", "IPTW-Multinomial", "IPTW-GBM", "IPTW-SL", "RAMS-Multinomial", "RAMS-GBM", "RAMS-SL".

formula	A formula object representing the variables used for the analysis. The default is to use all terms specified in <code>x</code> .
discard	A logical indicating whether to use the discarding rules for the BART based methods. The default is FALSE.
estimand	A character string representing the type of causal estimand. Only "ATT" or "ATE" is allowed. When the <code>estimand = "ATT"</code> , users also need to specify the reference treatment group by setting the <code>reference_trt</code> argument.
trim_perc	A numeric value indicating the percentile at which the inverse probability of treatment weights should be trimmed. The default is NULL.
SL.library	A character vector of prediction algorithms. A list of functions included in the SuperLearner package can be found with listWrappers .
reference_trt	A numeric value indicating reference treatment group for ATT effect.
boot	A logical indicating whether or not to use nonparametric bootstrap to calculate the 95% confidence intervals of the causal effect estimates. The default is FALSE.
nboots	A numeric value representing the number of bootstrap samples.
verbose_boot	A logical value indicating whether to print the progress of nonparametric bootstrap. The default is TRUE.
ndpost	A numeric value indicating the number of posterior draws for the Bayesian methods ("BART" and "RA").
caliper	A numeric value denoting the caliper which should be used when matching (<code>method = "VM"</code>) on the logit of GPS within each cluster formed by K-means clustering. The caliper is in standardized units. For example, <code>caliper = 0.25</code> means that all matches greater than 0.25 standard deviations of the logit of GPS are dropped. The default value is 0.25.
n_cluster	A numeric value denoting the number of clusters to form using K means clustering on the logit of GPS when <code>method = "VM"</code> . The default value is 5.
...	Other parameters that can be passed through to functions.

Value

A list of causal estimands including risk difference (RD), odds ratios (OR) and relative risk (RR) between different treatment groups.

References

Hu, L., Gu, C., Lopez, M., Ji, J., & Wisnivesky, J. (2020). Estimation of causal effects of multiple treatments in observational studies with a binary outcome. *Statistical Methods in Medical Research*, 29(11), 3218–3234.

Hu, L., Gu, C. Estimation of causal effects of multiple treatments in healthcare database studies with rare outcomes. *Health Service Outcomes Research Method* 21, 287–308 (2021).

Examples

```
lp_w_all <-
  c(".4*x1 + .1*x2 - .1*x4 + .1*x5", # w = 1
```

```

      ".2 * x1 + .2 * x2 - .2 * x4 - .3 * x5") # w = 2
nlp_w_all <-
  c("-.5*x1*x4 - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(300, 0, 0.5)",# x1
  "rbeta(300, 2, .4)", # x2
  "runif(300, 0, 0.5)",# x3
  "rweibull(300,1,2)", # x4
  "rbinom(300, 1, .4)"# x5
)
set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5,0,1.5),
  delta = c(0.5,0.5),
  psi = 1
)
ce_estimate(y = data$y, x = data$covariates, w = data$w,
  ndpost=100, method = "RA", estimand = "ATE")

```

data_sim

Simulate data for binary outcome with multiple treatments

Description

The function `data_sim` simulate data for binary outcome with multiple treatments. Users can adjust the following 7 design factors: (1) sample size, (2) ratio of units across treatment groups, (3) whether the treatment assignment model and the outcome generating model are linear or nonlinear, (4) whether the covariates that best predict the treatment also predict the outcome well, (5) whether the response surfaces are parallel across treatment groups, (6) outcome prevalence, and (7) degree of covariate overlap.

Usage

```

data_sim(
  sample_size,
  n_trt,
  X,
  lp_y,

```

```

    nlp_y,
    align = TRUE,
    tau,
    delta,
    psi,
    lp_w,
    nlp_w
  )

```

Arguments

sample_size	A numeric value indicating the total number of units.
n_trt	A numeric value indicating the number of treatments.
X	A vector of characters representing covariates, with each covariate being generated from the standard probability distributions in the stats package.
lp_y	A vector of characters of length n_trt, representing the linear effects in the outcome generating model.
nlp_y	A vector of characters of length n_trt, representing the nonlinear effects in the outcome generating model.
align	A logical indicating whether the predictors in the treatment assignment model are the same as the predictors for the outcome generating model. The default is TRUE. If the argument is set to FALSE, users need to specify additional two arguments lp_w and nlp_w.
tau	A numeric vector of length n_trt inducing different outcome event probabilities across treatment groups.
delta	A numeric vector of length n_trt-1 inducing different ratio of units across treatment groups.
psi	A numeric value for the parameter governing the sparsity of covariate overlap.
lp_w	is a vector of characters of length n_trt - 1, representing in the treatment assignment model
nlp_w	is a vector of characters of length n_trt - 1, representing in the treatment assignment model

Value

A list with 7 elements for simulated data. It contains

covariates:	X matrix
w:	treatment indicators
y:	observed binary outcomes
y_prev:	outcome prevalence rates
ratio_of_units:	the proportions of units in each treatment group
overlap_fig:	the visualization of covariate overlap via boxplots of the distributions of true GPS
Y_true:	simulated true outcome in each treatment group

References

Hu, L., Ji, J. (2021). CIMTx: An R package for causal inference with multiple treatments using observational data. arXiv:2110.10276

Examples

```
library(CIMTx)
lp_w_all <-
  c(".4*x1 + .1*x2 - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2 - .2 * x4 - .3 * x5") # w = 2
nlp_w_all <-
  c("-.5*x1*x4 - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(300, 0, 0.5)",# x1
  "rbeta(300, 2, .4)", # x2
  "runif(300, 0, 0.5)",# x3
  "rweibull(300,1,2)", # x4
  "rbinom(300, 1, .4)"# x5
)
set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5,0,1.5),
  delta = c(0.5,0.5),
  psi = 1
)
```

plot.CIMTx_IPTW

Boxplot for weight distribution

Description

This function make the boxplot plot for the weights estimated by different IPTW methods. The inputs of the function are from the output of `ce_estimate.R` function when the methods are "IPTW-Multinomial", "IPTW-GBM", "IPTW-SL".

Usage

```
## S3 method for class 'CIMTx_IPTW'
plot(...)
```

Arguments

... Objects from IPTW related methods

Value

A ggplot figure

Examples

```
iptw_object_example <- list(weight = rnorm(1000,1,1), method = "IPTW-SL")  
class(iptw_object_example) <- "CIMTx_IPTW"  
plot(iptw_object_example)
```

plot.CIMTx_sa_grid *Contour plot for the grid specification of sensitivity analysis*

Description

This function make the countor plot after the grid specification of sensitivity analysis. The input of the function is from the output of the sa.R function.

Usage

```
## S3 method for class 'CIMTx_sa_grid'  
plot(x, ATE = NULL, ATT = NULL, ...)
```

Arguments

x Object from sa function
ATE a character indicating the ATE effect to plot, eg, "1,3" or "2,3"
ATT a character indicating the ATT effect to plot, eg, "1,3" or "1,2"
... further arguments passed to or from other methods.

Value

A ggplot figure

Examples

```
sa_object_example <- list(ATE13 = seq(0,1,length.out = 25), grid_index = c(4,5),  
c_functions = data.frame(c4 = rep(seq(-0.6,0,0.15), each = 5),  
c5 = rep(seq(0,0.6,0.15), 5)))  
class(sa_object_example) <- "CIMTx_sa_grid"  
plot(sa_object_example, ATE = "1,3")
```

sa

*Flexible Monte Carlo sensitivity analysis for unmeasured confounding***Description**

The function `sa` implements the flexible sensitivity analysis approach for unmeasured confounding with multiple treatments and a binary outcome.

Usage

```
sa(
  x,
  y,
  w,
  formula = NULL,
  prior_c_function,
  M1,
  M2 = NULL,
  nCores = 1,
  estimand,
  reference_trt,
  ...
)
```

Arguments

<code>x</code>	A dataframe, including all the covariates but not treatments.
<code>y</code>	A numeric vector (0, 1) representing a binary outcome.
<code>w</code>	A numeric vector representing the treatment groups.
<code>formula</code>	A formula object for the analysis. The default is to use all terms specified in <code>x</code> .
<code>prior_c_function</code>	1) A vector of characters indicating the prior distributions for the confounding functions. Each character contains the random number generation code from the standard probability distributions in the <code>stats</code> package. 2) A vector of characters including the grid specifications for the confounding functions. It should be used when users want to formulate the confounding functions as scalar values. 3) A matrix indicating the point mass prior for the confounding functions
<code>M1</code>	A numeric value indicating the number of draws of the GPS from the posterior predictive distribution
<code>M2</code>	A numeric value indicating the number of draws from the prior distributions of the confounding functions
<code>nCores</code>	A numeric value indicating number of cores to use for parallel computing.

estimand A character string representing the type of causal estimand. Only "ATT" or "ATE" is allowed. When the estimand = "ATT", users also need to specify the reference treatment group by setting the reference_trt argument.

reference_trt A numeric value indicating reference treatment group for ATT effect.

... Other parameters that can be passed to BART functions

Value

A list of causal estimands including risk difference (RD) between different treatment groups.

References

Hu, L., Ji, J. (2021). CIMTx: An R package for causal inference with multiple treatments using observational data. arXiv:2110.10276

Examples

```
lp_w_all <-
  c(".4*x1 + .1*x2 - 1.1*x4 + 1.1*x5", # w = 1
    ".2 * x1 + .2 * x2 - 1.2 * x4 - 1.3 * x5") # w = 2
nlp_w_all <-
  c("-.5*x1*x4 - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(100, 0, 0.5)",# x1
  "rbeta(100, 2, .4)", # x2
  "runif(100, 0, 0.5)",# x3
  "rweibull(100,1,2)", # x4
  "rbinom(100, 1, .4)"# x5
)
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5,-0.5,0.5),
  delta = c(0.5,0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)",# c(1,2)
  "runif(0, 0.6)",# c(2,1)
  "runif(-0.6, 0)", # c(2,3)
```

```

"seq(-0.6, 0, by = 0.3)", # c(1,3)
"seq(0, 0.6, by = 0.3)", # c(3,1)
"runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_result <-
sa(
  M1 = 1,
  x = data$covariates,
  y = data$y,
  w = data$w,
  prior_c_function = c_grid,
  nCores = 1,
  estimand = "ATE",
)

```

```
summary.CIMTx_ATE_posterior
```

Summarize a CIMTx_ATE_posterior object

Description

Summarize a CIMTx_ATE_posterior object

Usage

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

Arguments

object a CIMTx_ATE_posterior object obtained with `ce_estimate` function.
 ... further arguments passed to or from other methods.

Value

a list with $w*(w-1)/2$ elements for ATE effect. Each element of the list contains the estimation, standard error, lower and upper 95% CI for RD/RR/OR.

Examples

```

library(CIMTx)
lp_w_all <-
c(".4*x1 + .1*x2 - .1*x4 + .1*x5", # w = 1
  ".2 * x1 + .2 * x2 - .2 * x4 - .3 * x5") # w = 2
nlp_w_all <-
c("-.5*x1*x4 - .1*x2*x5", # w = 1
  "-.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)

```

```

nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(300, 0, 0.5)",# x1
  "rbeta(300, 2, .4)", # x2
  "runif(300, 0, 0.5)",# x3
  "rweibull(300,1,2)", # x4
  "rbinom(300, 1, .4)"# x5
)
set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5,0,1.5),
  delta = c(0.5,0.5),
  psi = 1
)
ce_estimate_ra_ate_result <- ce_estimate(y = data$y, x = data$covariates
, w = data$w, ndpost = 10,method = "RA", estimand = "ATE")
summary(ce_estimate_ra_ate_result)

```

summary.CIMTx_ATE_sa *Summarize a CIMTx_ATE_sa object*

Description

Summarize a CIMTx_ATE_sa object

Usage

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

Arguments

object a CIMTx_ATE_sa object obtained with [sa](#) function.
 ... further arguments passed to or from other methods.

Value

a data frame containing the estimation, standard error, lower and upper 95% CI for the causal estimand in terms of RD.

Examples

```

lp_w_all <-
  c(".4*x1 + .1*x2 - 1.1*x4 + 1.1*x5", # w = 1
    ".2 * x1 + .2 * x2 - 1.2 * x4 - 1.3 * x5") # w = 2
nlp_w_all <-
  c("-.5*x1*x4 - .1*x2*x5", # w = 1
    " -.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(100, 0, 0.5)",# x1
  "rbeta(100, 2, .4)", # x2
  "runif(100, 0, 0.5)",# x3
  "rweibull(100,1,2)", # x4
  "rbinom(100, 1, .4)"# x5
)
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5,-0.5,0.5),
  delta = c(0.5,0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)",# c(1,2)
  "runif(0, 0.6)",# c(2,1)
  "runif(-0.6, 0)", # c(2,3)
  "seq(-0.6, 0, by = 0.3)", # c(1,3)
  "seq(0, 0.6, by = 0.3)", # c(3,1)
  "runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_ATE_result <-
sa(
  M1 = 1,
  x = data$covariates,
  y = data$y,
  w = data$w,
  prior_c_function = c_grid,
  nCores = 1,
  estimand = "ATE",
)
summary(sensitivity_analysis_parallel_ATE_result)

```

```
summary.CIMTx_ATT_posterior
      Summarize a CIMTx_ATT_posterior object
```

Description

Summarize a CIMTx_ATT_posterior object

Usage

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

Arguments

`object` a CIMTx_ATT_posterior object obtained with `ce_estimate` function.
`...` further arguments passed to or from other methods.

Value

a list with `w-1` elements for ATT effect. Each element of the list contains the estimation, standard error, lower and upper 95% CI for RD/RR/OR.

Examples

```
lp_w_all <-
  c(".4*x1 + .1*x2 - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2 - .2 * x4 - .3 * x5") # w = 2
nlp_w_all <-
  c("-.5*x1*x4 - .1*x2*x5", # w = 1
    " -.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
  "rnorm(300, 0, 0.5)",# x1
  "rbeta(300, 2, .4)", # x2
  "runif(300, 0, 0.5)",# x3
  "rweibull(300,1,2)", # x4
  "rbinom(300, 1, .4)"# x5
)
set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
```

```

nlp_w = nlp_w_all,
tau = c(-1.5,0,1.5),
delta = c(0.5,0.5),
psi = 1
)
ce_estimate_ra_att_result <- ce_estimate(y = data$y, x = data$covariates ,
w = data$w, reference_trt = 1, ndpost = 10, method = "RA", estimand = "ATT")
summary(ce_estimate_ra_att_result)

```

summary.CIMTx_ATT_sa *Summarize a CIMTx_ATT_sa object*

Description

Summarize a CIMTx_ATT_sa object

Usage

```

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

```

Arguments

object a CIMTx_ATT_sa object obtained with [sa](#) function.
... further arguments passed to or from other methods.

Value

a data frame containing the estimation, standard error, lower and upper 95% CI for the causal estimand in terms of RD.

Examples

```

lp_w_all <-
c(".4*x1 + .1*x2 - 1.1*x4 + 1.1*x5", # w = 1
".2 * x1 + .2 * x2 - 1.2 * x4 - 1.3 * x5") # w = 2
nlp_w_all <-
c("-.5*x1*x4 - .1*x2*x5", # w = 1
"-.3*x1*x4 + .2*x2*x5")# w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1 - .1*x2*x3", 3)
X_all <- c(
"rnorm(100, 0, 0.5)",# x1
"rbeta(100, 2, .4)", # x2
"runif(100, 0, 0.5)",# x3
"rweibull(100,1,2)", # x4
"rbinom(100, 1, .4)"# x5
)

```

```
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  X = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5, -0.5, 0.5),
  delta = c(0.5, 0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)", # c(1,2)
  "runif(0, 0.6)", # c(2,1)
  "runif(-0.6, 0)", # c(2,3)
  "seq(-0.6, 0, by = 0.3)", # c(1,3)
  "seq(0, 0.6, by = 0.3)", # c(3,1)
  "runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_ATT_result <-
sa(
  M1 = 1,
  x = data$covariates,
  y = data$y,
  w = data$w,
  prior_c_function = c_grid,
  nCores = 1,
  estimand = "ATE",
)
summary(sensitivity_analysis_parallel_ATT_result)
```

Index

`ce_estimate`, [2](#), [10](#), [13](#)

`data_sim`, [4](#)

`distributions`, [5](#), [8](#)

`formula`, [3](#), [8](#)

`listWrappers`, [3](#)

`plot.CIMTx_IPTW`, [6](#)

`plot.CIMTx_sa_grid`, [7](#)

`sa`, [8](#), [11](#), [14](#)

`stats`, [5](#), [8](#)

`summary.CIMTx_ATE_posterior`, [10](#)

`summary.CIMTx_ATE_sa`, [11](#)

`summary.CIMTx_ATT_posterior`, [13](#)

`summary.CIMTx_ATT_sa`, [14](#)