Package 'SAMTx'

January 20, 2025

Type Package

Title Sensitivity Assessment to Unmeasured Confounding with Multiple
Treatments

Version 0.3.0

Description A sensitivity analysis approach for unmeasured confounding in observa-
tional data with multiple treatments and a binary outcome. This approach derives the gen-
eral bias formula and provides adjusted causal effect estimates in response to various assump-
tions about the degree of unmeasured confounding. Nested multiple imputation is embed-
ded within the Bayesian framework to integrate uncertainty about the sensitivity parame-
ters and sampling variability. Bayesian Additive Regression Model (BART) is used for out-
come modeling. The causal estimands are the conditional average treatment ef-
fects (CATE) based on the risk difference. For more details, see paper: Hu L et al. (2020) A flex-
ible sensitivity analysis approach for unmeasured confounding with multiple treatments and a bi-
nary outcome with application to SEER-Medicare lung cancer data <arXiv:2012.06093>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports BART

NeedsCompilation no

Author Liangyuan Hu [aut],
Jungang Zou [aut],
Jiayi Ji [aut, cre]

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

Repository CRAN

Date/Publication 2021-06-28 15:10:02 UTC

Contents

| sensitivity_analysis | *Sensitivity Assessment to Unmeasured Confounding with Multiple Treatments* |

Description

This function implements the nested multiple imputation using Bayesian Additive Regression Trees (BART)

Usage

```
sensitivity_analysis(
  covariates,
  y,
  A,
  alpha,
  n_p,
  nposterior = 1000,
  sensitivity_correction = TRUE
)
```

Arguments

covariates	Dataframe including all the covariates
y	Numeric vector for the binary outcome
A	Numeric vector for the treatment indicator
alpha	Priors for sensitivity parameters
n_p	Number of nested imputations to conduct
nposterior	Number of posterior samples, default is 1000
sensitivity_correction	
	Whether to use sensitivity correction algorithm, default is TRUE

Value

A list of dataframes for each ATE between different treatments. If number of treatments = 3, it contains

ATE12:	A dataframe with number of rows = n_p * nrow(alpha) and number of columns = length(y)
ATE23:	A dataframe with number of rows = n_p * nrow(alpha) and number of columns = length(y)
ATE13:	A dataframe with number of rows = n_p * nrow(alpha) and number of columns = length(y)

Examples

```
sample_size = 10
x1 = rbinom(sample_size, 1, prob=0.4)
x2 = rbinom(sample_size, 1, prob=0.5)
lp.A = 0.2 * x1 + 0.4 * x2 + rnorm(sample_size, 0, 0.1)
lp.B = -0.3 * x1 + 0.8 * x2 + rnorm(sample_size, 0, 0.1)
lp.C = 0.1 * x1 + 0.5 * x2 + rnorm(sample_size, 0, 0.1)
# calculate the true probability of assignment
p.A1 <- exp(lp.A)/(exp(lp.A)+exp(lp.B)+exp(lp.C))
p.A2 <- exp(lp.B)/(exp(lp.A)+exp(lp.B)+exp(lp.C))
p.A3 <- exp(lp.C)/(exp(lp.A)+exp(lp.B)+exp(lp.C))
p.A <- matrix(c(p.A1,p.A2,p.A3),ncol = 3)
A = NULL
for (m in 1:sample_size) { # assign treatment
 A[m] <- sample(c(1, 2, 3),
                size = 1,
                replace = TRUE,
                prob = p.A[m, ])
}
table(A)
# set the binary outcome
Y2 = 0.3 * x1 + 0.2 * x1 * x2 + 1.3 * x2
Y1 = -0.6 * x1 + 0.5 * x2 + 0.3 * x1 * x2
Y0 = -0.8 * x1 - 1.2 * x2 + 1.5 * x2 * x1
Y2 = rbinom(sample_size, 1, exp(Y2)/(1+exp(Y2)))
Y1 = rbinom(sample_size, 1, exp(Y1)/(1+exp(Y1)))
Y0 = rbinom(sample_size, 1, exp(Y0)/(1+exp(Y0)))
dat = cbind(Y0, Y1, Y2, A)
Yobs <- apply(dat, 1, function(x)
 x[1:3][x[4]]) #observed when trt is received
n = 1
alpha = cbind(
 runif(n, mean(Y0[A ==1])-mean(Y0[A ==2]) - 0.001, mean(Y0[A ==1])-mean(Y0[A ==2]) + 0.001),
 runif(n, mean(Y1[A ==2])-mean(Y1[A ==1]) - 0.001, mean(Y1[A ==2])-mean(Y1[A ==1]) + 0.001),
 runif(n, mean(Y1[A ==2])-mean(Y1[A ==3]) - 0.001, mean(Y1[A ==2])-mean(Y1[A ==3]) + 0.001),
 runif(n, mean(Y0[A ==1])-mean(Y0[A ==3]) - 0.001, mean(Y0[A ==1])-mean(Y0[A ==3]) + 0.001),
 runif(n, mean(Y2[A ==3])-mean(Y2[A ==1]) - 0.001, mean(Y2[A ==3])-mean(Y2[A ==1]) + 0.001),
 runif(n, mean(Y2[A ==3])-mean(Y2[A ==2]) - 0.001, mean(Y2[A ==3])-mean(Y2[A ==2]) + 0.001))
y <- Yobs
n_p <- 1
sample_gap <- 10
sensitivity_analysis_result <- sensitivity_analysis(cbind(x1, x2), Yobs,
A, alpha, n_p = 1, sensitivity_correction = TRUE)
mean(sensitivity_analysis_result$ATE_12)
mean(sensitivity_analysis_result$ATE_02)
mean(sensitivity_analysis_result$ATE_01)
```

Index