Package 'SBdecomp'

March 25, 2020

Type Package

Title Quantifies the Proportion of the Estimated Selection Bias
Explained by Observed Confounders

Version 1.0

Date 2020-03-25

Author Layla Parast

Maintainer Layla Parast <parast@rand.org>

Description Uses parametric and nonparametric methods to quantify the proportion of the estimated selection bias explained by each observed confounder when estimating propensity score weighted treatment effects.

License GPL

Imports stats, twang, graphics, survey

NeedsCompilation no

Depends R (>= 3.5.0)

R topics documented:

`SBdecomp-package` *Quantifies the Proportion of the Estimated Selection Bias Explained by Observed Confounders*

Description

Uses parametric and nonparametric methods to quantify the proportion of the estimated selection bias explained by each observed confounder when estimating propensity score weighted treatment effects.

Details

The DESCRIPTION file:

Package:	SBdecomp
Type:	Package
Title:	Quantifies the Proportion of the Estimated Selection Bias Explained by Observed Confounders
Version:	1.0
Date:	2020-03-25
Author:	Layla Parast
Maintainer:	Layla Parast <parast@rand.org>
Description:	Uses parametric and nonparametric methods to quantify the proportion of the estimated selection bias expl
License:	GPL
Imports:	stats, twang, graphics, survey

Index of help topics:

```
SBdecomp-package        Quantifies the Proportion of the Estimated
                        Selection Bias Explained by Observed
                        Confounders
bar.sbdecomp            Creates a Bar Plot
petsdata                Dog ownership dataset
sbdecomp                Selection Bias Decomposition
```

This packge provides a function that decomposes the estimated selection bias to quantify what proportion of the estimated selection bias is explained by each observed confounder used in the propensity score model; the function is sbdecomp. The function offers two approaches - confounder inclusion or removal, and offers two estimation approaches - parametric or nonparametric. These methods allow one to identify the most important confounder when estimating a propensity score weighted treatment effect in the presence of selection bias.

Author(s)

Layla Parast

Maintainer: Layla Parast <parast@rand.org>

References

Parast, L and Griffin, BA (2019). Quantifying the Bias due to Observed Individual Confounders in Causal Treatment Effect Estimates, Under Review.

Examples

```
data(petsdata)
sbdecomp(outcome = petsdata$genhealth, treatment = petsdata$gotdog, confounders =
as.data.frame(petsdata[,c(2:13)]), type = "inclusion", estimation = "parametric")

sbdecomp(outcome = petsdata$genhealth, treatment = petsdata$gotdog, confounders =
as.data.frame(petsdata[,c(2:13)]), type = "inclusion", estimation = "parametric",
Bonly =FALSE, balance = TRUE)

sbdecomp(outcome = "genhealth", treatment = "gotdog", confounders = c("age",
"ismale", "race_coll","hhsize","ownhome", "married",  "ontanf", "hhincome",
"fulltime","spouse_fulltime" ,"liveinhouse", "ruralurban"), data = petsdata,
type = "inclusion", estimation = "parametric", Bonly =FALSE, balance = TRUE)
```

`bar.sbdecomp` *Creates a Bar Plot*

Description

This function creates a bar plot to illustrate the estimated proportion of the observed selection bias explained by each observed confounder used in the propensity score model.

Usage

```
bar.sbdecomp(output.list, main)
```

Arguments

`output.list`	List outputed from main sbdecomp function
`main`	Title for plot, optional

Value

A bar plot showing the estimated proportions of the estimated selection bias explained by the observed confounders used in the propensity score model is created.

Author(s)

Layla Parast

References

Parast, L and Griffin, BA (2019). Quantifying the Bias due to Observed Individual Confounders in Causal Treatment Effect Estimates, Under Review.

Examples

```
data(petsdata)
output = sbdecomp(outcome = petsdata$genhealth, treatment = petsdata$gotdog, confounders =
as.data.frame(petsdata[,c(2:13)]), type = "inclusion", estimation = "parametric",
balance = FALSE)

bar.sbdecomp(output)
```

`petsdata` *Dog ownership dataset*

Description

This dataset is a random subset of the publicly available 2003 California Health Interview Survey data; the data consists responses from 2,102 adults. For the purpose of illustrating the functions in this package, the goal is to investigate the effect of dog ownership on general health. Dog ownership was assessed with the question “Do you have any dogs that you allow inside your home?"; 29.0% of respondents owned a dog. General health status of the individual was measured as the self-reported response to the question “Would you say that in general your health is excellent, very good, good, fair or poor?" Responses were coded from 1 through 5 with 5 indicating “Excellent." Available individual characteristics i.e., confounders, in this dataset include age, gender, race/ethnicity, household size, marriage status, whether the individual received TANF (Temporary Assistance for Needy Families), household annual income, whether the individual worked full time, whether the individual had a spouse that worked full time, whether the individual lived in a house, and a rural/urban measure (1= urban; 2= 2nd city; 3 = suburban; 4 = town and rural) for the individual’s address.

Usage

```
data("petsdata")
```

Format

A data frame with 2102 observations on the following 14 variables.

`gotdog` indicates whether individual reported owning a dog where 1 is yes and 0 is no (binary)

`age` age of the individual (numeric)

`ismale` indicates the sex of the individual where 1 is yes and 0 is no (binary)

`race_coll` ace/ethnicity of the individual, collapsed into categories where 1 is Latino, 4 is Asian, 6 is White, and 7 is a combined category of Pacific Islander/Native American/African American/other race/multiple race (factor/categorical)

`hhsize` household size (numeric count)

`ownhome` indicates whether the individual owns the home they live in where 1 is yes and 0 is no (binary)

`married` indicates whether the individual is married where 1 is yes and 0 is no (binary)

`ontanf` indicates whether the individual received TANF where 1 is yes and 0 is no (binary)

`hhincome` household income, log-transformed (numeric)

`fulltime` indicates whether the individual works full time where 1 is yes and 0 is no (binary)

`spouse_fulltime` indicates whether the individual has a spouse that works full time where 1 is yes and 0 is no (binary)

`liveinhouse` indicates whether the individual lives in a house (as opposed to an apartment, duplex, or mobile home) where 1 is yes and 0 is no (binary)

`ruralurban` the rural/urban measure of the home address of the individual where 1= urban, 2= 2nd city, 3 = suburban, 4 = town and rural (numeric)

`genhealth` self-reported general health of the individual, numeric 1-5 scale where 5 is excellent (numeric)

References

California Health Interview Survey (2005). CHIS 2003 Methodology Series. Los Angeles: UCLA Center for Health Policy Research.

Examples

```
data(petsdata)
dim(petsdata)
names(petsdata)
```

---

sbdecomp *Selection Bias Decomposition*

Description

This function decomposes the estimated selection bias to quantify what proportion of the estimated selection bias is explained by each observed confounder used in the propensity score model when estimating propensity score weighted treatment effects. The function offers two approaches - confounder inclusion or removal, and offers two estimation approaches - parametric or nonparametric.

Usage

```
sbdecomp(outcome, treatment, confounders, data=NULL, type = "inclusion", estimation
= "parametric", Bonly = T, balance = T, n.trees = 20000, interaction.depth = 4,
shrinkage = 0.005, verbose = FALSE, stop.method = c("es.max"), cv.folds = 0,
standard.error = F, boot.rep=500)
```

Arguments

`outcome` Continuous outcome; numeric vector or name of the variable in supplied data

`treatment` Binary treatment indicator, 0 or 1; numeric vector or name of the variable in supplied data

`confounders` a data frame of confounders with names or a vector containing the names of the confounders in supplied data (each confounder must be either numeric or factor)

`data` data containing all variables; required if names provided in outcome, treatment, and/or confounders arguments above

`type` The type of method for selection bias deocomposition; options are inclusion or removal, default is inclusion

`estimation` The type of estimation that should be used; options are parametric (based on logistic regression) or nonparametric (based on generalized boosted models [GBM]), default is parametric

`Bonly` TRUE or FALSE; whether the user only wants the quantities B which are the proportions of the selection bias explained by each confounder, default is TRUE

`balance` TRUE or FALSE; whether the user wants balance information, default is TRUE

`n.trees` The n.trees for the nonparametric GBM approach, if using nonparametric estimation; default is 20000

`interaction.depth`
The interaction depth for the nonparametric GBM approach, if using nonparametric estimation; default is 4

`shrinkage` The shrinkage for the nonparametric GBM approach, if using nonparametric estimation; default is 0.005

`verbose` TRUE or FALSE for the nonparametric GBM approach, if using nonparametric estimation; default is FALSE

`stop.method` The stopping method for the nonparametric GBM approach, if using nonparametric estimation; default is "es.max"

`cv.folds` The cross-validation folds for the nonparametric GBM approach, if using nonparametric estimation; default is 0

`standard.error` TRUE or FALSE; whether the user wants standard error estimates for the resulting estimates, obtained using boostrapping; default is FALSE; note that if using nonparametric estimation, obtaining the standard error estimates will be very time-intensive

`boot.rep` The number of bootstrap replications to be used when calculating the standard error; default is 500

Details

Details can be found in Parast, L and Griffin, BA (2019). Quantifying the Bias due to Observed Individual Confounders in Causal Treatment Effect Estimates, Under Review. Please email parast@rand.org for a current version.

Value

A list is returned:

`delta.naive` Estimated naive treatment effect

`p.value.delta.naive` P-value for estimated naive treatment effect

`delta.fully.adjusted` Estimated treatment effect after adjusting for all variables

`p.value.delta.fully.adjusted` P-value for estimated treatment effect after adjusting for all variables

`B` The proportion of the selection bias explained by each variable

`estimated.selection.bias` The estimated selection bias, only if Bonly = FALSE

`lambda` The defined selection bias movement when including/removing a variable, only if Bonly = FALSE

`delta.each` The adjusted delta including only each variable or dropping each variable (depending on type), only if Bonly = FALSE

`balance.naive.mean` mean of the standardized effect size differences across all variables between the two groups; only if balance = TRUE

`balance.naive.max` max of the standardized effect size differences across all variables between the two groups; only if balance = TRUE

`balance.fully.adjusted.mean` mean of the standardized effect size differences across all variables between the two groups, after weighting with propensity score weights calculated using all variables; only if balance = TRUE

`balance.fully.adjusted.max`
: max of the standardized effect size differences across all variables between the two groups, after weighting with propensity score weights calculated using all variables; only if balance = TRUE

`balance.mean`
: mean of the standardized effect size differences across all variables in the relevant propensity score model between the two groups, after weighting with relevant propensity score weights; if using type = "inclusion" and variable included is a factor, output will be the mean effect size difference across all levels; only if balance = TRUE

`balance.max`
: max of the standardized effect size differences across all variables in the relevant propensity score model between the two groups, after weighting with relevant propensity score weights; if using type = "inclusion" and variable included is a factor, output will be the max effect size difference across all levels; only if balance = TRUE

`B.standard.error`
: The estimated standard error for the estimated proportion of the selection bias explained by each variable; only if standard.error = TRUE

`lambda.standard.error`
: The estimated standard error for the estimated selection bias movement when including/removing a variable; only if standard.error = TRUE

Author(s)

Layla Parast

References

Parast, L and Griffin, BA (2019). Quantifying the Bias due to Observed Individual Confounders in Causal Treatment Effect Estimates, Under Review.

Examples

```
data(petsdata)
sbdecomp(outcome = petsdata$genhealth, treatment = petsdata$gotdog, confounders =
as.data.frame(petsdata[,c(2:13)]), type = "inclusion", estimation = "parametric",
balance = FALSE)

sbdecomp(outcome = petsdata$genhealth, treatment = petsdata$gotdog, confounders =
as.data.frame(petsdata[,c(2:13)]), type = "inclusion", estimation = "parametric",
Bonly =FALSE, balance = TRUE)

sbdecomp(outcome = "genhealth", treatment = "gotdog", confounders = c("age",
"ismale", "race_coll","hhsize","ownhome", "married",  "ontanf", "hhincome",
"fulltime","spouse_fulltime" ,"liveinhouse", "ruralurban"), data = petsdata,
type = "inclusion", estimation = "parametric", Bonly =FALSE, balance = TRUE)
```

Index