Package: CausalGAM (via r-universe)

September 3, 2024

Version 0.1-4

Date 2017-10-16

Title Estimation of Causal Effects with Generalized Additive Models

Author Adam Glynn <adam.glynn@emory.edu>, Kevin Quinn <kmq@umich.edu>

Maintainer Kevin Quinn <kmq@umich.edu>

Depends R (>= 2.9.0), gam (>= 1.0.1)

Description Implements various estimators for average treatment effects - an inverse probability weighted (IPW) estimator, an augmented inverse probability weighted (AIPW) estimator, and a standard regression estimator - that make use of generalized additive models for the treatment assignment model and/or outcome model. See: Glynn, Adam N. and Kevin M. Quinn. 2010. ``An Introduction to the Augmented Inverse Propensity Weighted Estimator." Political Analysis. 18: 36-56.

License GPL-2

NeedsCompilation no

Date/Publication 2017-10-19 19:56:10 UTC

Repository https://quinn-kevin.r-universe.dev

RemoteUrl https://github.com/cran/CausalGAM

RemoteRef HEAD

RemoteSha 2243d0866d25f694a0f76e56a2a7f7b62017583d

Contents

balance.IPW *Check Post-Weighting Balance for (A)IPW Estimators Using Generalized Additive Models*

Description

This function calculates weighted means of covariates where weights in inverse propensity weights and then examines the differences in the weighted means across treated and control units as a diagnostic for covariate balance.

Usage

```
balance.IPW(pscore.formula, pscore.family,
            treatment.var, outcome.var, data = NULL,
            divby0.action = c("fail", "truncate", "discard"),
            divby0.tol = 1e-08, nboot = 501,
            suppress.warnings = TRUE, ...)
```

Arguments

pscore.formula	A formula expression for the propensity score model. See the documentation of gam for details.
pscore.family	A description of the error distribution and link function to be used for the propensity score model. See the documentation of gam for details.
treatment.var	A character variable giving the name of the binary treatment variable in data. If treatment.var is a numeric variable, it is assumed that *control* corresponds to sort(unique(treatment.values))[1] and *treatment* corresponds to sort(unique(treatment.value If treatment.var is a factor, it is assumed that *control* corresponds to levels(treatment.values)[1] and *treatment* corresponds to levels(treatment.values)[2].
outcome.var	A character variable giving the name of the outcome variable in data.
data	A *non-optional* data frame containing the variables in the propensity score model along with all covariates that one wishes to assess balance for. data **cannot contain any missing values.**
divby0.action	A character variable describing what action to take when some estimated propensity scores are less than divby0.tol or greater than $1 -$ divby0.tol. Options include: 'fail' (abort the call to estimate.ATE), 'truncate' (set all estimated propensity scores less than divby0.tol equal to divby0.tol and all estimated propensity scores greater than $1 -$ divby0.tol equal to $1 -$ divby0.tol), and 'discard' (discard units that have estimate propensity scores less than divby0.tol or greater than $1 -$ divby0.tol). Note that discarding units will change the estimand.
divby0.tol	A scalar in $[0, 0.5)$ giving the tolerance level for extreme propensity scores. Defaults to $1e - 8$. See divby0.action for details.
nboot	Number of bootrap replications used for calculating bootstrap standard errors. If nboot is less than or equal to 0 then bootstrap standard errors are not calculated. Defaults to 501.

`suppress.warnings`
: Logical value indicating whether warnings from the `gam` fitting procedures should be suppressed from printing to the screen. Defaults to `TRUE`.

`...`
: Further arguments to be passed.

Details

This function provides diagnostic information that allows a user to judge whether the inverse propensity weights generated from a particular generalized additive model specification result in covariate balance across treated and control groups. The function is intended to be used before the `estimate.ATE` function in order to find a specification for the propensity score model that results in sufficient covariate balance.

The weighted mean differences between all variables in the dataset passed to `balance.IPW` are reported along with a z-statistics for these weighted differences. Univariate mean covariate balance is decreasing in the absolute value of the z-statistics (z-statistics closer to 0 imply better univariate mean balance).

Printing the output from `balance.IPW` will result in a table with $k - 2$ rows (one for each variable other than the treatment and outcome variables) and 6 columns. The columns are (from left to right) the observed mean of the covariate among the treated units, the observed mean of the covariate among the control units, the weighted mean of the covariate among the treated units, the weighted mean of the covariate among the control units, the weighted mean difference, and the z-statistic for the difference.

It is often useful to include interactions and powers of the covariates in the dataset so that balance can be checked for these quantities as well.

Means, mean differences, and z-statistics are only reported for numeric covariates.

Value

An object of class `balance` with the following attributes:

`obs.mean.control`
: The observed mean of each of the covariates within the control units.

`obs.mean.treated`
: The observed mean of each of the covariates within the treated units.

`weighted.mean.control`
: The weighted mean of each of the covariates within the control units.

`weighted.mean.treated`
: The weighted mean of each of the covariates within the treated units.

`weighted.diff.SE`
: The bootstrap standard errors for the differences betwen `weighted.mean.treated` and `weighted.mean.control`.

Author(s)

Adam Glynn, Emory University

Kevin Quinn, University of Michigan

References

Adam N. Glynn and Kevin M. Quinn. 2010. "An Introduction to the Augmented Inverse Propensity Weighted Estimator." *Political Analysis*.

See Also

gam, estimate.ATE

Examples

```
## Not run:
set.seed(1234)
## number of units in sample
n <- 2000


## measured potential confounders
z1 <- rnorm(n)
z2 <- rnorm(n)
z3 <- rnorm(n)
z4 <- rnorm(n)

## treatment assignment
prob.treated <- pnorm(-0.5 + 0.75*z2)
x <- rbinom(n, 1, prob.treated)

## potential outcomes
y0 <- z4 + rnorm(n)
y1 <- z1 + z2 + z3 + cos(z3*2) + rnorm(n)

## observed outcomes
y <- y0
y[x==1] <- y1[x==1]


## put everything in a data frame
examp.data <- data.frame(z1, z2, z3, z4, x, y)

## augment data with interactions and powers of covariates
examp.data$z1z1 <- examp.data$z1^2
examp.data$z2z2 <- examp.data$z2^2
examp.data$z3z3 <- examp.data$z3^2
examp.data$z4z4 <- examp.data$z4^2

examp.data$z1z2 <- examp.data$z1 * examp.data$z2
examp.data$z1z3 <- examp.data$z1 * examp.data$z3
examp.data$z1z4 <- examp.data$z1 * examp.data$z4

examp.data$z2z3 <- examp.data$z2 * examp.data$z3
examp.data$z2z4 <- examp.data$z2 * examp.data$z4

examp.data$z3z4 <- examp.data$z3 * examp.data$z4
```

```
## check balance of a propensity score model that is not sufficient to
## control confounding bias

bal.1 <- balance.IPW(pscore.formula=x~s(z3)+s(z4),
                     pscore.family=binomial(probit),
                     treatment.var="x",
                     outcome.var="y",
                     data=examp.data,
                     nboot=250)

print(bal.1) ## some big z-statistics here indicating balance not so great


## try again
bal.2 <- balance.IPW(pscore.formula=x~z1+z2+z3+z4,
                     pscore.family=binomial(probit),
                     treatment.var="x",
                     outcome.var="y",
                     data=examp.data,
                     nboot=250)

print(bal.2) ## balance looks much better--
             ##    only 1 out of 14 zs > 2.0 in absval


## End(Not run)
```

`estimate.ATE` *Estimate Population Average Treatment Effects (ATE) Using Generalized Additive Models*

Description

This function implements three estimators for the population ATE— a regression estimator, an inverse propensity weighted (IPW) estimator, and an augmented inverse propensity weighted (AIPW) estimator— using generalized additive models.

Usage

```
estimate.ATE(pscore.formula, pscore.family,
             outcome.formula.t, outcome.formula.c, outcome.family,
             treatment.var, data = NULL,
             divby0.action = c("fail", "truncate", "discard"),
             divby0.tol = 1e-08, nboot = 501,
             variance.smooth.deg = 1, variance.smooth.span = 0.75,
             var.gam.plot = TRUE, suppress.warnings = TRUE, ...)
```

Arguments

`pscore.formula` A formula expression for the propensity score model. See the documentation of `gam` for details.

`pscore.family` A description of the error distribution and link function to be used for the propensity score model. See the documentation of `gam` for details.

`outcome.formula.t`
A formula expression for the outcome model under active treatment. See the documentation of `gam` for details.

`outcome.formula.c`
A formula expression for the outcome model under control. See the documentation of `gam` for details.

`outcome.family` A description of the error distribution and link function to be used for the outcome models. See the documentation of `gam` for details.

`treatment.var` A character variable giving the name of the binary treatment variable in `data`. If `treatment.var` is a numeric variable, it is assumed that *control* corresponds to `sort(unique(treatment.values))[1]` and *treatment* corresponds to `sort(unique(treatment.value` If `treatment.var` is a factor, it is assumed that *control* corresponds to `levels(treatment.values)[1]` and *treatment* corresponds to `levels(treatment.values)[2]`.

`data` A *non-optional* data frame containing the variables in the model. `data` **cannot contain any missing values.**

`divby0.action` A character variable describing what action to take when some estimated propensity scores are less than `divby0.tol` or greater than $1 -$ `divby0.tol`. Options include: '`fail`' (abort the call to `estimate.ATE`), '`truncate`' (set all estimated propensity scores less than `divby0.tol` equal to `divby0.tol` and all estimated propensity scores greater than $1 -$ `divby0.tol` equal to $1 -$ `divby0.tol`), and '`discard`' (discard units that have estimate propensity scores less than `divby0.tol` or greater than $1 -$ `divby0.tol`). Note that discarding units will change the estimand.

`divby0.tol` A scalar in $[0, 0.5)$ giving the tolerance level for extreme propensity scores. Defaults to $1e - 8$. See `divby0.action` for details.

`nboot` Number of bootrap replications used for calculating bootstrap standard errors. If `nboot` is less than or equal to 0 then bootstrap standard errors are not calculated. Defaults to 501.

`variance.smooth.deg`
The degree of the loess smooth used to calculate the conditional error variance of the outcome models given the estimated propensity scores. Possible values are 0, 1, or 2. Defaults to 1. If set to a value less than 0 than the conditional error variance will not be calculated and the estimated asymptotic standard errors will not be reported. See `lo` for details.

`variance.smooth.span`
The span of the loess smooth used to calculate the conditional error variance of the outcome models given the estimated propensity scores. Defaults to 0.75. If set to a value less than or equal to 0 than the conditional error variance will not be calculated and the estimated asymptotic standard errors will not be reported.See `lo` for details.

`var.gam.plot` Logical value indicating whether the estimated conditional variances should be plotted against the estimated propensity scores. Setting `var.gam.plot` to `TRUE` is useful for judging whether `variance.smooth.deg` and `variance.smooth.span` were set appropriately. Defaults to `TRUE`.

`suppress.warnings` Logical value indicating whether warnings from the `gam` fitting procedures should be suppressed from printing to the screen. Defaults to `TRUE`.

`...` Further arguments to be passed.

Details

The three estimators implemented by this function are a regression estimator, an IPW estimator with weights normalized to sum to 1, and an AIPW estimator. Glynn and Quinn (2010) provides details regarding how each of these estimators are implemented. The AIPW estimator requires the specification of both a propensity score model governing treatment assignment and outcome models that describe the conditional expectation of the outcome variable given measured confounders and treatment status. The AIPW estimator has the so-called double robustness property. This means that if either the propensity score model or the outcomes models are correctly specified then the estimator is consistent for ATE.

Standard errors for the regression and IPW estimators can be calculated by either the bootstrap or by estimating the large sample standard errors. The latter approach requires estimation of the conditional variance of the disturbances in the outcome models given the propensity scores (see section IV of Imbens (2004) for details). The accuracy of these standard errors is only as good as one's estimates of these conditional variances.

Standard errors for the AIPW estimator can be calculated similarly. In addition, Lunceford and Davidian (2004) also discuss an empirical sandwich estimator of the sampling variance which is also implemented here.

Value

An object of class `CausalGAM` with the following attributes:

`ATE.AIPW.hat` AIPW estimate of ATE.

`ATE.reg.hat` Regression estimate of ATE.

`ATE.IPW.hat` IPW estimate of ATE.

`ATE.AIPWsand.SE` Empirical sandwich standard error for `ATE.AIPW.hat`.

`ATE.AIPW.asymp.SE` Estimated asymptotic standard error for `ATE.AIPW.hat`.

`ATE.reg.asymp.SE` Estimated asymptotic standard error for `ATE.reg.hat`.

`ATE.IPW.asymp.SE` Estimated asymptotic standard error for `ATE.IPW.hat`.

`ATE.AIPW.bs.SE` Estimated bootstrap standard error for `ATE.AIPW.hat`.

`ATE.reg.bs.SE` Estimated bootstrap standard error for `ATE.reg.hat`.

`ATE.IPW.bs.SE` Estimated bootstrap standard error for `ATE.IPW.hat`.

`ATE.AIPW.bs` Vector of bootstrap replications of `ATE.AIPW.hat`.

`ATE.reg.bs` Vector of bootstrap replications of `ATE.reg.hat`.

`ATE.IPW.bs` Vector of bootstrap replications of `ATE.IPW.hat`.

`gam.t` `gam` object from fitted outcome model under treatment.

`gam.c` `gam` object from the fitted outcome model under control.

`gam.ps` `gam` object from the fitted propensity score model.

`truncated.indic`
Logical vector indicating which rows of `data` had extreme propensity scores truncated.

`discarded.indic`
Logical vector indicating which rows of `data` were discarded because of extreme propensity scores.

`treated.value` Value of `treatment.var` that corresponds to active treatment.

`control.value` Value of `treatment.var` that corresponds to control.

`treatment.var` `treatment.var`

`n.treated.prediscard`
Number of treated units before any truncations or discards.

`n.control.prediscard`
Number of control units before any truncations or discards.

`n.treated.postdiscard`
Number of treated units after truncations or discards.

`n.control.postdiscard`
Number of control units after truncations or discards.

`pscores.prediscard`
Estimated propensity scores before any truncations or discards.

`pscores.postdiscard`
Estimated propensity scores after truncations or discards.

`cond.var.t` Vector of conditional error variances for the outcome for each unit under treatment given the unit's estimated propensity score.

`cond.var.c` Vector of conditional error variances for the outcome for each unit under control given the unit's estimated propensity score.

`call` The initial call to `estimate.ATE`.

`data` The data frame sent to `estimate.ATE`.

Author(s)

Adam Glynn, Emory University

Kevin Quinn, University of Michigan

References

Adam N. Glynn and Kevin M. Quinn. 2010. "An Introduction to the Augmented Inverse Propensity Weighted Estimator." *Political Analysis*.

Guido W. Imbens. 2004. "Nonparametric Estimation of Average Treatment Effects Under Exogeneity: A Review." *The Review of Economics and Statistics*. 86: 4-29.

Jared K. Lunceford and Marie Davidian. 2004. "Stratification and Weighting via the Propensity Score in Estimation of Causal Treatment Effects: A Comparative Study." *Statistics in Medicine*. 23: 2937-2960.

See Also

gam, balance.IPW

Examples

```
## Not run:
## a simulated data example with Gaussian outcomes
##

## number of units in sample
n <- 2000


## measured potential confounders
z1 <- rnorm(n)
z2 <- rnorm(n)
z3 <- rnorm(n)
z4 <- rnorm(n)

## treatment assignment
prob.treated <- pnorm(-0.5 + 0.75*z2)
x <- rbinom(n, 1, prob.treated)

## potential outcomes
y0 <- z4 + rnorm(n)
y1 <- z1 + z2 + z3 + cos(z3*2) + rnorm(n)

## observed outcomes
y <- y0
y[x==1] <- y1[x==1]


## put everything in a data frame
examp.data <- data.frame(z1, z2, z3, z4, x, y)


## estimate ATE
##
## in a real example one would want to use a larger number of
## bootstrap replications
##
```

```
ATE.out <- estimate.ATE(pscore.formula = x ~ s(z2),
                        pscore.family = binomial,
                        outcome.formula.t = y ~ s(z1) + s(z2) + s(z3) + s(z4),
                        outcome.formula.c = y ~ s(z1) + s(z2) + s(z3) + s(z4),
         outcome.family = gaussian,
   treatment.var = "x",
                        data=examp.data,
                        divby0.action="t",
                        divby0.tol=0.001,
                        var.gam.plot=FALSE,
   nboot=50)


## print summary of estimates
print(ATE.out)




## a simulated data example with Bernoulli outcomes
##

## number of units in sample
n <- 2000


## measured potential confounders
z1 <- rnorm(n)
z2 <- rnorm(n)
z3 <- rnorm(n)
z4 <- rnorm(n)

## treatment assignment
prob.treated <- pnorm(-0.5 + 0.75*z2)
x <- rbinom(n, 1, prob.treated)

## potential outcomes
p0 <- pnorm(z4)
p1 <- pnorm(z1 + z2 + z3 + cos(z3*2))
y0 <- rbinom(n, 1, p0)
y1 <- rbinom(n, 1, p1)

## observed outcomes
y <- y0
y[x==1] <- y1[x==1]


## put everything in a data frame
examp.data <- data.frame(z1, z2, z3, z4, x, y)


## estimate ATE
##
```

```
## in a real example one would want to use a larger number of
## bootstrap replications
##
ATE.out <- estimate.ATE(pscore.formula = x ~ s(z2),
                        pscore.family = binomial,
                        outcome.formula.t = y ~ s(z1) + s(z2) + s(z3) + s(z4),
                        outcome.formula.c = y ~ s(z1) + s(z2) + s(z3) + s(z4),
         outcome.family = binomial,
   treatment.var = "x",
                        data=examp.data,
                        divby0.action="t",
                        divby0.tol=0.001,
                        var.gam.plot=FALSE,
   nboot=50)


## print summary of estimates
print(ATE.out)

## End(Not run)
```

Index