Package 'npcausal'

February 14, 2021

Type Package

Title Nonparametric causal inference methods

Version 0.1.0

Author Edward H. Kennedy

Maintainer Edward H. Kennedy <edward@stat.cmu.edu>

Description This package provides a variety of tools for nonparametric estimation of causal effects across a wide range of settings. The methods are based on the theory of influence functions, and can incorporate flexible machine learning and high-dimensional regression tools, while still yielding inference in the form of confidence intervals and hypothesis tests. Many of the methods are doubly robust.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

R topics documented:

ate *Doubly robust estimation of average treatment effect*

Description

ate is used to estimate the mean outcome in a population had all subjects received given levels of a discrete (unconfounded) treatment, using doubly robust methods with ensembled nuisance estimation.

Usage

```
ate(y, a, x, nsplits=2, sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet",
  "SL.glm.interaction", "SL.mean","SL.ranger","rpart"))
```

Arguments

y	outcome of interest.
a	discrete treatment.
x	covariate matrix.
nsplits	integer number of sample splits for nuisance estimation. If nsplits=1, sample splitting is not used, and nuisance functions are estimated on full sample (in which case validity of SEs/CIs requires empirical process conditions). Otherwise must have nsplits>1.
sl.lib	algorithm library for SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", "ranger", "rpart.

Value

A list containing the following components:

res	estimates/SEs/CIs/p-values for population means and relevant contrasts.
nuis	subject-specific estimates of nuisance functions (i.e., propensity score and outcome regression)
ifvals	matrix of estimated influence function values.

References

Robins JM, Rotnitzky A (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*.

Hahn J (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*.

van der Laan MJ, Robins JM (2003). *Unified Methods for Censored Longitudinal Data and Causality* (Springer).

Tsiatis AA (2006). *Semiparametric Theory and Missing Data* (Springer).

Robins JM, Li L, Tchetgen Tchetgen ET, van der Vaart A (2008). Higher order influence functions and minimax estimation of nonlinear functionals. *Probability and Statistics: Essays in Honor of David A. Freedman*.

Zheng W, van der Laan (2010). Asymptotic theory for cross-validated targeted maximum likelihood estimation *UC Berkeley Division of Biostatistics Working Paper Series*.

Chernozhukov V, Chetverikov V, Demirer M, et al (2016). Double machine learning for treatment and causal parameters.

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
a <- sample(3,n,replace=TRUE); y <- rnorm(n,mean=x[,1])

ate.res <- ate(y,a,x, sl.lib=c("SL.mean","SL.gam"))
```

---

att *Estimating average effect of treatment on the treated*

Description

`att` is used to estimate the difference in mean outcome among treated subjects had a binary (unconfounded) treatment been withheld.

Usage

```
att(y, a, x, nsplits=2, sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet",
  "SL.glm.interaction","SL.mean","SL.ranger"))
```

Arguments

`y`	outcome of interest.
`a`	binary treatment.
`x`	covariate matrix.
`nsplits`	integer number of sample splits for nuisance estimation. If nsplits=1, sample splitting is not used, and nuisance functions are estimated on full sample (in which case validity of SEs/CIs requires empirical process conditions). Otherwise must have nsplits>1.
`sl.lib`	algorithm library if using SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", and "ranger".

Value

A list containing the following components:

`res`	estimates/SEs/CIs/p-values for treated means and contrast.
`nuis`	subject-specific estimates of nuisance functions (i.e., propensity score and outcome regression)
`ifvals`	vector of estimated influence function values.

References

(Also see references for function `ate`)

Kennedy EH, Sjolander A, Small DS (2015). Semiparametric causal inference in matched cohort studies. *Biometrika*.

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
a <- rbinom(n,1,.3); y <- rnorm(n)

att.res <- att(y,a,x)
```

`cdensity` *Doubly robust series estimation of counterfactual densities*

Description

`cdensity` is used to estimate counterfactual densities, i.e., the density of the potential outcome in a population if everyone received given treatment levels, using doubly robust estimates of L2 projections of the density onto a linear basis expansion. Nuisance functions are estimated with random forests. The L2 distance between the density of the counterfactuals is also estimated as a density-based treatment effect.

Usage

```
cdensity(y, a, x, kmax=5, l2 = TRUE,
 gridlen=20, nsplits=2, progress_updates = TRUE,
 makeplot=TRUE, kforplot=5, ylim=NULL)
```

Arguments

`y`	outcome of interest.
`a`	binary treatment (more than 2 levels are allowed, but only densities under A=1 and A=0 will be estimated).
`x`	covariate matrix.
`kmax`	Integer indicating maximum dimension of (cosine) basis expansion that should be used in series estimator.
`l2`	A `logical` value indicating whether an estimate of the L2 distance between counterfactual densities (under A=1 vs A=0) should be returned.
`gridlen`	Integer number indicating length of grid for which the plug-in estimator of the marginal density is computed.
`nsplits`	Integer number of sample splits for nuisance estimation. If `nsplits = 1`, sample splitting is not used, and nuisance functions are estimated n full sample (in which case validity of standard errors and confidence intervals requires empirical process conditions). Otherwise must have `nsplits > 1`.
`progress_updates`	A `logical` value indicating whether to print a progress statement as various stages of computation reach completion. The default is `TRUE`, printing a progress bar to inform the user.
`makeplot`	A `logical` value indicating whether to print a plot.
`kforplot`	A vector of two integers indicating which k values to plot results for, with first argument for A=1 and second for A=0.
`ylim`	Range of y values at which density should be plotted.

Value

A list containing the following components:

res	estimates/SEs/CIs/p-values for population means and relevant contrasts.
nuis	subject-specific estimates of nuisance functions (i.e., propensity score and outcome regression)
ifvals	matrix of estimated influence function values.

References

Kennedy EH, Wasserman LA, Balakrishnan S. Semiparametric counterfactual density estimation. arxiv:TBA

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
a <- sample(3,n,replace=TRUE)-2; y <- rnorm(n)

cdens.res <- cdensity(y,a,x)
```

---

ctseff *Estimating average effect curve for continuous treatment*

Description

ctseff is used to estimate the mean outcomes in a population had all subjects received given levels of a continuous (unconfounded) treatment.

Usage

```
ctseff(y, a, x, bw.seq, sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet",
  "SL.glm.interaction","SL.mean","SL.ranger"))
```

Arguments

y	outcome of interest.
a	continuous treatment.
x	covariate matrix.
bw.seq	sequence of bandwidth values.
sl.lib	algorithm library for SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", and "ranger".

Value

A list containing the following components:

res	estimates/SEs/CIs for population means.
bw.risk	estimated risk at sequence of bandwidth values.

References

Kennedy EH, Ma Z, McHugh MD, Small DS (2017). Nonparametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society, Series B*. arxiv:1507.00747

Examples

```
n <- 500
x <- matrix(rnorm(n * 5), nrow = n)
a <- runif(n)
y <- a + rnorm(n, sd = .5)

ce.res <- ctseff(y, a, x, bw.seq = seq(.2, 2, length.out = 100))
plot.ctseff(ce.res)

# check that bandwidth choice is minimizer
plot(ce.res$bw.risk$bw, ce.res$bw.risk$risk)
```

`cv.cdensity` *Cross-validation for doubly robust estimation of counterfactual densities*

Description

`cv.cdensity` estimates counterfactual densities using linear cosine basis expansions at a sequence of dimensions, and then estimates the L2 pseudo-risk of each, which can be used for purposes of model selection. Nuisance functions are estimated with random forests.

Usage

```
cv.cdensity(y, a, x, kmax=5,
 gridlen=20, nsplits=2, progress_updates = TRUE)
```

Arguments

`y`	outcome of interest.
`a`	binary treatment (more than 2 levels are allowed, but only densities under A=1 and A=0 will be estimated).
`x`	covariate matrix.
`kmax`	Integer indicating maximum dimension of (cosine) basis expansion that should be used in series estimator.
`gridlen`	Integer number indicating length of grid for which the plug-in estimator of the marginal density is computed.
`nsplits`	Integer number of sample splits for nuisance estimation. If `nsplits = 1`, sample splitting is not used, and nuisance functions are estimated n full sample (in which case validity of standard errors and confidence intervals requires empirical process conditions). Otherwise must have `nsplits > 1`.
`progress_updates`	A `logical` value indicating whether to print a progress statement as various stages of computation reach completion. The default is `TRUE`, printing a progress bar to inform the user.

Value

A plot of the pseudo L2 risk of candidate estimators for counterfactual densities, at each model dimension from 1 to kmax

References

Kennedy EH, Wasserman LA, Balakrishnan S. Semiparametric counterfactual density estimation. arxiv:TBA

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
a <- sample(3,n,replace=TRUE)-2; y <- rnorm(n)

cv.cdensity(y,a,x)
```

---

ipsi *Estimating effects of incremental propensity score interventions*

Description

`ipsi` is used to estimate effects of incremental propensity score interventions, i.e., estimates of mean outcomes if the odds of receiving treatment were multiplied by a factor delta.

Usage

```
ipsi(y, a, x.trt, x.out, time, id, delta.seq, nsplits, ci_level = 0.95,
 progress_bar = TRUE, return_ifvals = FALSE, fit,
 sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet","SL.glm.interaction", "SL.mean","SL.ranger","
```

Arguments

`y`	Outcome of interest measured at end of study.
`a`	Binary treatment.
`x.trt`	Covariate matrix for treatment regression.
`x.out`	Covariate matrix for outcome regression.
`time`	Measurement time.
`id`	Subject identifier.
`delta.seq`	Sequence of delta increment values for incremental propensity score intervention.
`nsplits`	Integer number of sample splits for nuisance estimation. If `nsplits = 1`, sample splitting is not used, and nuisance functions are estimated n full sample (in which case validity of standard errors and confidence intervals requires empirical process conditions). Otherwise must have `nsplits > 1`.
`ci_level`	A `numeric` value giving the level (1 - alpha) of the confidence interval to be computed around the point estimate.

progress_bar	A logical value indicating whether to print a customized progress bar as various stages of computation reach completion. The default is TRUE, printing a progress bar to inform the user.
return_ifvals	A logical indicating whether the estimated observation-level values of the influence function ought to be returned as part of the output object. The default is FALSE as these values are rarely of interest in standard usage.
fit	How nuisance functions should be estimated. Options are "rf" for random forests via the ranger package, or "sl" for super learner.
sl.lib	sl.lib algorithm library for SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", "ranger", "rpart.

Value

A list containing the following components:

res	estimates/SEs and uniform CIs for population means.
res.ptwise	estimates/SEs and pointwise CIs for population means.
calpha	multiplier bootstrap critical value.

Details

Treatment and covariates are expected to be time-varying and measured throughout the course of the study. Therefore if n is the number of subjects and T the number of timepoints, then a, time, and id should all be vectors of length nxT, and x.trt and x.out should be matrices with nxT rows. However y should be a vector of length n since it is only measured at the end of the study. The subject ordering should be consistent across function inputs, based on the ordering specified by id. See example below for an illustration.

References

Kennedy EH. Nonparametric causal effects based on incremental propensity score interventions. arxiv:1704.00211

Examples

```
n <- 500
T <- 4

time <- rep(1:T, n)
id <- rep(1:n, rep(T, n))
x.trt <- matrix(rnorm(n * T * 5), nrow = n * T)
x.out <- matrix(rnorm(n * T * 5), nrow = n * T)
a <- rbinom(n * T, 1, .5)
y <- rnorm(mean=1,n)

d.seq <- seq(0.1, 5, length.out = 10)

ipsi.res <- ipsi(y, a, x.trt, x.out, time, id, d.seq)
```

ivbds *Estimating bounds on treatment effects with instrumental variables*

Description

ivbds is used to estimate bounds on various effects using instrumental variables.

Usage

```
ivbds(y, a, z, x, nsplits=2, sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet",
  "SL.glm.interaction", "SL.mean","SL.ranger","rpart"), project01=T)
```

Arguments

y	outcome of interest.
a	binary treatment.
z	binary instrument.
x	covariate matrix.
nsplits	integer number of sample splits for nuisance estimation. If nsplits=1, sample splitting is not used, and nuisance functions are estimated on full sample (in which case validity of SEs/CIs requires empirical process conditions). Otherwise must have nsplits>1.
sl.lib	algorithm library for SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", "ranger", "rpart".
project01	should the estimated compliance score be projected to space respecting 0-1 bounds and monotonicity?

Value

A list containing the following components:

res	estimates/SEs/CIs/p-values for local average treatment effect E(Y(a=1)-Y(a=0)|A(z=1)>A(z=0)), as well as IV strength and sharpness.
nuis	subject-specific estimates of nuisance functions (i.e., IV propensity score and treatment/outcome regressions)
ifvals	matrix of estimated influence function values.

References

(Also see references for function ate)

Angrist JD, Imbens GW, Rubin DB (1996). Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*.

Abadie A (2003). Semiparametric instrumental variable estimation of treatment response models. *Journal of Econometrics*.

Kennedy EH, Balakrishnan S, G'Sell M (2017). Complier classification with sharp instrumental variables. *Working Paper*.

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
z <- rbinom(n,1,0.5); a <- rbinom(n,1,0.6*z+0.2)
y <- rnorm(n)

ivbds.res <- ivbds(y,a,z,x)
```

`ivlate` *Estimating complier average effect of binary treatment using binary instrument*

Description

`ivlate` is used to estimate the mean outcome among compliers (i.e., those encouraged by the instrument) had all subjects received treatment versus control.

Usage

```
ivlate(y, a, z, x, nsplits=2, sl.lib=c("SL.earth","SL.gam","SL.glm","SL.glmnet",
  "SL.glm.interaction", "SL.mean","SL.ranger","rpart"), project01=T)
```

Arguments

`y`	outcome of interest.
`a`	binary treatment.
`z`	binary instrument.
`x`	covariate matrix.
`nsplits`	integer number of sample splits for nuisance estimation. If nsplits=1, sample splitting is not used, and nuisance functions are estimated on full sample (in which case validity of SEs/CIs requires empirical process conditions). Otherwise must have nsplits>1.
`sl.lib`	algorithm library for SuperLearner. Default library includes "earth", "gam", "glm", "glmnet", "glm.interaction", "mean", "ranger", "rpart".
`project01`	should the estimated compliance score be projected to space respecting 0-1 bounds and monotonicity?

Value

A list containing the following components:

`res`	estimates/SEs/CIs/p-values for local average treatment effect E(Y(a=1)-Y(a=0)|A(z=1)>A(z=0)), as well as IV strength and sharpness.
`nuis`	subject-specific estimates of nuisance functions (i.e., IV propensity score and treatment/outcome regressions)
`ifvals`	matrix of estimated influence function values.

References

(Also see references for function ate)

Angrist JD, Imbens GW, Rubin DB (1996). Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*.

Abadie A (2003). Semiparametric instrumental variable estimation of treatment response models. *Journal of Econometrics*.

Kennedy EH, Balakrishnan S, G'Sell M (2017). Complier classification with sharp instrumental variables. *Working Paper*.

Examples

```
n <- 100; x <- matrix(rnorm(n*5),nrow=n)
z <- rbinom(n,1,0.5); a <- rbinom(n,1,0.6*z+0.2)
y <- rnorm(n)

ivlate.res <- ivlate(y,a,z,x)
```

---

plot.cteff *Plot estimated average effect curve for continuous treatment*

Description

plot.cteff is used to plot results from cteff fit.

Usage

```
plot.cteff(cteff.res)
```

Arguments

cteff.res	output from cteff fit.

Value

A plot of estimated effect curve with pointwise confidence intervals.

References

Kennedy EH, Ma Z, McHugh MD, Small DS (2017). Nonparametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society, Series B*. arxiv:1507.00747

Examples

```
n <- 500; x <- matrix(rnorm(n*5),nrow=n)
a <- runif(n); y <- a + rnorm(n,sd=.5)

ce.res <- cteff(y,a,x, bw.seq=seq(.2,2,length.out=100))
plot.cteff(ce.res)
```

SL.ranger *Add Ranger wrapper for SuperLearner*

Description

SL.ranger is a wrapper for SuperLearner that adds the fast random forests method ranger.

Usage

```
SL.ranger(Y, X, newX, family, ...)
```

Arguments

Y	outcome vector.
X	covariate dataframe for training.
newX	covariate dataframe for predictions.
family	link function (currently only supports "gaussian" identity link).

Value

Predictions and fits from ranger.

References

Wright MN, Ziegler A (2016). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*.

Index