Package 'SDRcausal'

June 30, 2020

Version 0.3.0

Date 2020-06-11

Title SDRcausal

Author Filip Edstrom [aut, cre]

Maintainer Mohammad Ghasempour <mohammad.ghasempour@umu.se>

Description Provides two semiparametric estimators, imp.ate and ipw.ate.

Encoding UTF-8

Imports stats, ggplot2

Suggests nloptr

LazyData true

License GPL (>= 2)

RoxygenNote 7.1.0

NeedsCompilation yes

R topics documented:

aipw.ate . . . 2
aipw.var . . . 3
aipw2.ate . . . 4
b10_fun . . . 5
b_fun . . . 6
cms.ps.semi . . . 6
cms.semi . . . 8
example_data . . . 10
imp.ate . . . 10
imp.val . . . 13
imp.var . . . 14
imp2.var . . . 16
ipw.ate . . . 17
ipw.var . . . 19
nw_kernel_regress . . . 21
plot_imp . . . 22
plot_ipw . . . 23
ps.semi . . . 23

Index **26**

aipw.ate *Combines IPW and IMP estimators to form the augmented IPW, AIPW*

Description

Augmented IPW (AIPW) as in Ghosh, Ma, & De Luna (2020).

Usage

```
aipw.ate(y, treated, imp, ipw)
```

Arguments

y	Observed response
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()
ipw	ipw_output object from ipw.ate()

Value

Average treatment effect (ATE) for the augmented IPW (AIPW)

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 8,
           bwc_prop_score = 8)

# Calculate the Augmented IPW (AIPW)
aipw <- SDRcausal::aipw.ate(y, trt, imp, ipw)
```

aipw.var *Estimates Augmented Inverse Probability variance*

Description

Variance of the Augmented IPW as in Ghosh, Ma, & De Luna (2020).

Usage

```
aipw.var(
  x,
  y,
  treated,
  imp,
  ipw,
  bandwidth_scale1,
  bandwidth_scale0,
  bandwidth_scale_pr,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001,
  num_deriv_h = 1e-08,
  verbose = FALSE
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()
ipw	ipw_output object from ipw.ate()
bandwidth_scale1	Scaling of the calculated bandwidth, m1
bandwidth_scale0	Scaling of the calculated bandwidth, m0
bandwidth_scale_pr	Scaling of the calculated bandwidth, pr
kernel	Specifies which kernel function to be used
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/5).
gauss_cutoff	Cutoff value for Gaussian kernel
num_deriv_h	Step size of numerical derivative.
verbose	Specifies if the program should print output while running.

Value

The variance of Augmented IPW

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 10,
           bwc_prop_score = 18)

# Calculate the variance of the Augmented IPW (AIPW)
var <- SDRcausal::aipw.var(x, y, trt, imp, ipw,
           bandwidth_scale1 = imp$bw1, bandwidth_scale0 = imp$bw0,
           bandwidth_scale_pr = ipw$bw_pr)
```

---

aipw2.ate *Improved Augmented IPW (IAIPW)*

Description

Combines IPW and IMP estimators to form the improved augmented IPW, IAIPW as in Ghosh, Ma, & De Luna (2020).

Usage

```
aipw2.ate(y, treated, imp, ipw)
```

Arguments

y	Observed response
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()
ipw	ipw_output object from ipw.ate()

Value

Average treatment effect (ATE) for the improved augmented IPW (IAIPW)

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 10,
           bwc_prop_score = 18)

# Calculate the Improved Augmented IPW (AIPW2)
iaipw <- SDRcausal::aipw2.ate(y, trt, imp, ipw)
```

---

b10_fun *Calculates B1/0*

Description

Calculates Eq 2.8 or 2.10 in Ghosh, Ma, & De Luna (2020).

Usage

```
b10_fun(x, treated, dm, beta, kernel, bandwidth, gauss_cutoff)
```

Arguments

x	Projection of covariate matrix on CMS
treated	Binary vector indicating treatment.
dm	Derivative of imputed values
beta	CMS
kernel	Specifies which kernel function to be used
bandwidth	Specifies if bandwidth_scale will be used as the
gauss_cutoff	Cutoff value for Gaussian kernel

Value

B1/0 matrix

`b_fun` *Calculates B1/0*

Description

Calculates Eq 2.8 or 2.10 in Ghosh, Ma, & De Luna (2020).

Usage

```
b_fun(x, treated, alpha_hat, h, kernel, bandwidth, bandwidth_pr, verbose)
```

Arguments

`x`	Projection of covariate matrix on CMS
`treated`	Treated
`alpha_hat`	Derivative of imputed values
`h`	CMS
`kernel`	Specifies which kernel function to be used
`bandwidth`	Kernel bandwidth
`bandwidth_pr`	Kernel bandwidth for probability
`verbose`	Specifies if the program should print output while running.

Value

B1/0 matrix

`cms.ps.semi` *Estimates the Central Mean Space (CMS)*

Description

Semiparametric estimation of the Central Mean Space (CMS) as in Ghosh, Ma, & De Luna (2020). To be used with SDRcausal::ps.semi().

Usage

```
cms.ps.semi(
  x,
  treated,
  alpha_initial,
  solver = "optim",
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  bandwidth_scale = 1,
  gauss_cutoff = 0.001,
  penalty = 10,
  n_before_pen = 5,
  root_tol = 0.001,
  n_threads = 1,
  verbose = FALSE,
  ...
)
```

Arguments

x	Covariate matrix
treated	Binary vetor indicating treatment
alpha_initial	Initial guess of CMS
solver	Specifies which solver to be used. Current options optim and cobyla (from nloptr package).
kernel	Specifies which kernel function to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used explicitly as the bandwidth.
bandwidth_scale	Scaling of the calculated bandwidth, or in case of explicit_bandwidth = TRUE the bandwidth.
gauss_cutoff	cutoff value for Gaussian kernel
penalty	Penalty for the optimizer if a probability is outside (0, 1). Added to the function value in optim as: penalty^(n), where n is the number of probabilities outside (0, 1).
n_before_pen	Number of probabilities outside the range (0, 1) to accept during dimension reduction.
root_tol	Tolerance which makes the program warn if optim stops at at a value higher than root_tol.
n_threads	Sets number of threads for parallel run. Set to 0 serial. If n_threads exceeds maximum number of threads, sets n_threads to max_threads - 1. To use max_threads, set to n_threads to max_threads of system.
verbose	Specifies if the program should print output while running.
...	Additional parameters passed to solver.

Value

A list containing the final alpha, bandwwidth used, and the output of optim

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

See Also

[stats::optim]

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
alp <- SDRcausal::alpha_guess

# Perform semiparametric dimension reduction for treated
cms <- SDRcausal::cms.ps.semi(x, trt, alp,
           explicit_bandwidth = TRUE, bandwidth_scale = 8)
```

`cms.semi` *Estimates the Central Mean Space (CMS)*

Description

Semiparametric estimation of the Central Mean Space (CMS) as in Ghosh, Ma, & De Luna (2020). To be used with SDRcausal::imp.val().

Usage

```
cms.semi(
  x,
  y,
  treated,
  beta_initial,
  solver = "optim",
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  bandwidth_scale = 1,
  gauss_cutoff = 0.001,
  penalty = 10,
  n_before_pen = 1,
  root_tol = 0.001,
  n_threads = 1,
  verbose = FALSE,
  ...
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vetor indicating treatment
beta_initial	Initial guess of CMS
solver	Specifies which solver to be used. Current options optim and cobyla (from nloptr package).
kernel	Specifies which kernel function to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/5)
bandwidth_scale	Scaling of the bandwidth or the actual bandwidth if explicit bandwidth.
gauss_cutoff	cutoff value for Gaussian kernel
penalty	Penalty for the optimizer if local linear regression fails. Added to the function value in solver as: penalty^(n - n_before_pen), where n is the number of llr fails.
n_before_pen	Number of probabilities outside the range (0, 1) to accept during dimension reduction.
root_tol	Tolerance which makes the program warn if optim stops at at a value higher than root_tol.
n_threads	Sets number of threads for parallel run. Set to 0 serial. If n_threads exceeds maximum number of threads, sets n_threads to max_threads - 1. To use max_threads, set to n_threads to max_threads of system.
verbose	Specifies if the program should print output while running.
...	Additional parameters passed to optim.

Value

A list containing the final beta, the bandwidth used, a warning if optim does not converge or converges to a value that is larger than root_tol, and the output of optim.

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

See Also

[stats::optim]

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
```

```
trt1 <- SDRcausal::treated
trt0 <- rep(1, length(trt1)) - trt1
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess

# Perform semiparametric dimension reduction for treated
cms1 <- SDRcausal::cms.semi(x, y, trt1, b1,
           explicit_bandwidth = TRUE, bandwidth_scale = 1)

# Perform semiparametric dimension reduction for untreated
cms0 <- SDRcausal::cms.semi(x, y, trt0, b0,
           explicit_bandwidth = TRUE, bandwidth_scale = 1)
```

`example_data` *Example data*

Description

Data generated as in paper, study 1. Using the betas in betas data. Use beta1/0 for imputation as the initial guess of the central mean space (CMS) and alpha as the initial guess of the CMS for IPW.

Format

Data used in examples of the SDRcausal package

- covariatescovariate matrix
- outcomesobserved outcome vector
- treatedbinary treatment vector
- beta1_guessStarting guess for CMS for treated
- beta0_guessStarting guess for CMS for untreated
- alpha_guessStarting guess for CMS for propensity score

`imp.ate` *Estimates Average Treatment Effect (ATE) by imputation (IMP)*

Description

Semiparametric estimation of the average treatment effect based on the imputation method described in Ghosh, Ma, & De Luna (2020).

Usage

```
imp.ate(
  x,
  y,
  treated1,
  beta_guess1,
  beta_guess0,
  solver = "optim",
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  recalc_bandwidth = FALSE,
  bwc_dim_red1 = 1,
  bwc_impute1 = 1,
  bwc_dim_red0 = 1,
  bwc_impute0 = 1,
  gauss_cutoff = 0.001,
  penalty = 10,
  n_before_pen = 5,
  to_extrapolate = TRUE,
  to_truncate = TRUE,
  extrapolation_basis = 5,
  n_threads = 1,
  verbose = TRUE,
  ...
)
```

Arguments

x	Covariate matrix
y	Response vector
treated1	Binary vector indicating treatment.
beta_guess1	Initial guess of beta for m1
beta_guess0	Initial guess of beta for m0
solver	Specifies which solver to be used. Current options optim and cobyla (from nloptr package).
kernel	Specifies which kernel function to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
recalc_bandwidth	Specifies wheter the bandwidth should be recalculated after the estimation of alpha (cms.ps.semi).
bwc_dim_red1	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the banddwidth. For dimension reduction (cms.semi).
bwc_impute1	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the banddwidth. Recalculated if explicit_bandwidth = FALSE and recalc_bandwidth = TRUE. For imputation.
bwc_dim_red0	See bwc_dim_red1

bwc_impute0	See bwc_impute1
gauss_cutoff	Cutoff value for Gaussian kernel
penalty	Penalty for the optimizer if local linear regression fails. Added to the function value in solver as: penalty^(n - n_before_pen), where n is the number of llr fails.
n_before_pen	Number of probabilities outside the range (0, 1) to accept during dimension reduction.
to_extrapolate	Specifies wheter to extrapolate or not
to_truncate	Specifies wheter to extrapolate or not
extrapolation_basis	Number of data point to base extrapolation on.
n_threads	Sets number of threads for parallel run. Set to 0 serial. If n_threads exceeds maximum number of threads, sets n_threads to max_threads - 1. To use max_threads, set to n_threads to max_threads of system.
verbose	Specifies if the program should print output while running.
...	Additional parameters passed to optim.

Value

A list containing the average treatment effect of the combination of observed and imputed values (ate), the average treatment effect based on the imputed values only (ate2), the imputed values for treated (m1) and untreated treated (m0), the and the output from optim (op).

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

See Also

[stats::optim]

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)
```

`imp.val` *Estimates imputed values based on CMS*

Description

Performs semiparametric imputation based on the CMS calculated by cms.semi, as in Ghosh, Ma, & De Luna (2020).

Usage

```
imp.val(
  x,
  y,
  treated,
  beta_hat,
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  bandwidth_scale = 1,
  gauss_cutoff = 0.001,
  to_extrapolate = TRUE,
  to_truncate = TRUE,
  extrapolation_basis = as.integer(5),
  verbose = FALSE
)
```

Arguments

`x`	Covariate matrix
`y`	Response vector
`treated`	Binary vetor indicating treatment
`beta_hat`	Locally efficient CMS
`kernel`	Specifies which kernel function to be used
`explicit_bandwidth`	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3)
`bandwidth_scale`	Kernel bandwidth
`gauss_cutoff`	Cutoff value for Gaussian kernel
`to_extrapolate`	Specifies wheter to extrapolate or not
`to_truncate`	Specifies wheter to extrapolate or not
`extrapolation_basis`	Number of data point to base extrapolation on.
`verbose`	Specifies if the program should print output while running

Value

A list containing the reduced space xb, the imputed values and their derivatives.

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess

# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt1 <- SDRcausal::treated
n <- as.integer(dim(x)[1])
trt0 <- as.integer(rep(1, times = n) - trt1)
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess

# Perform semiparametric dimension reduction for treated
cms1 <- SDRcausal::cms.semi(x, y, trt1, b1,
           explicit_bandwidth = 1, bandwidth_scale = 1)

# Perform semiparametric dimension reduction for untreated
cms0 <- SDRcausal::cms.semi(x, y, trt0, b0,
           explicit_bandwidth = 1, bandwidth_scale = 1)

# Perform semiparametric imputation for treated
m1 <- SDRcausal::imp.val(x, y, trt1, cms1$fb,
         explicit_bandwidth = 1, bandwidth_scale = cms1$bw)

# Perform semiparametric imputation for untreated
m0 <- SDRcausal::imp.val(x, y, trt0, cms0$fb,
         explicit_bandwidth = 1, bandwidth_scale = cms0$bw)
```

`imp.var` *Estimates IMP variance*

Description

Variance of the IMP as in Ghosh, Ma, & De Luna (2020).

Usage

```
imp.var(
  x,
  y,
  treated,
  imp,
  ipw,
  bandwidth_scale1,
  bandwidth_scale0,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()
ipw	ipw_output object from ipw.ate()
bandwidth_scale1	Scaling of the calculated bandwidth, or in case explicit_bandwidth the actual bandwidth. For m1 and beta1.
bandwidth_scale0	See bandwidth_scale1. For m0 and beta0.
kernel	Specifies which kernel function to be used
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
gauss_cutoff	Cutoff value for Gaussian kernel

Value

Variance of IMP

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
```

```
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 10,
           bwc_prop_score = 18)

# Calculate the variance of the Augmented IPW (AIPW)
var <- SDRcausal::imp.var(x, y, trt, imp, ipw,
           bandwidth_scale1 = imp$bw1, bandwidth_scale0 = imp$bw0)
```

`imp2.var` *Estimates IMP2 variance*

Description

Variance of IMP2 as in Ghosh, Ma, & De Luna (2020).

Usage

```
imp2.var(
  x,
  y,
  treated,
  imp,
  ipw,
  bandwidth_scale1,
  bandwidth_scale0,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001
)
```

Arguments

`x`	Covariate matrix
`y`	Response vector
`treated`	Binary vetor indicating treatment
`imp`	imp_output object from imp.ate()
`ipw`	ipw_output object from ipw.ate()
`bandwidth_scale1`	Scaling of the calculated bandwidth, or in case explicit_bandwidth the actual bandwidth. For m1 and beta1.
`bandwidth_scale0`	See bandwidth_scale1. For m0 and beta0.

kernel	Specifies which kernel function to be used
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
gauss_cutoff	Cutoff value for Gaussian kernel

Value

Variance of IMP

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 10,
           bwc_prop_score = 18)

# Calculate the variance of the Augmented IPW (AIPW)
var <- SDRcausal::imp2.var(x, y, trt, imp, ipw,
           bandwidth_scale1 = imp$bw1, bandwidth_scale0 = imp$bw0)
```

---

ipw.ate *Estimates average treatment effect through IPW*

Description

Semiparametric estimation of the average treatment effect based on the IPW method described in Ghosh, Ma, & De Luna (2020).

Usage

```
ipw.ate(
  x,
  y,
  treated,
  alpha_initial,
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  recalc_bandwidth = TRUE,
  bwc_dim_red = 1,
  bwc_prop_score = 10,
  gauss_cutoff = 0.001,
  penalty = 10,
  n_before_pen = 1,
  n_threads = 1,
  verbose = TRUE,
  ...
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vector indicating treatment.
alpha_initial	Initial guess of beta for m1
kernel	Specifies which kernel function to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
recalc_bandwidth	Specifies wheter the bandwidth should be recalculated after the estimation of alpha (cms.ps.semi)
bwc_dim_red	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the banddwidth. For dimension reduction (cms.ps.semi).
bwc_prop_score	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the banddwidth. Recalculated if explicit_bandwidth = FALSE and recalc_bandwidth = TRUE. For propensity score.
gauss_cutoff	cutoff value for Gaussian kernel
penalty	Penalty for the optimizer if a probability is outside (0, 1) during dimension reduction. Added to the function value in solver as: penalty^(n - n_before_pen), where n is the number of probabilities outside (0, 1).
n_before_pen	Number of probabilities outside the range (0, 1) to accept during dimension reduction.
n_threads	Sets number of threads for parallel run. Set to 0 serial. If n_threads exceeds maximum number of threads, sets n_threads to max_threads - 1. To use max_threads, set to n_threads to max_threads of system.
verbose	Specifies if the program should print output while running.
...	Additional parameters passed to optim.

Value

A list containing the average treatment effect (ate), the propensity score (pr), the final alpha (fa), and the output from optim (op).

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

See Also

[stats::optim]

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
alp <- SDRcausal::alpha_guess

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 8,
           bwc_prop_score = 8)
```

---

ipw.var *Estimates IPW variance*

Description

Variance of the IPW as in Ghosh, Ma, & De Luna (2020).

Usage

```
ipw.var(
  x,
  y,
  treated,
  imp,
  ipw,
  bandwidth_scale,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001,
  num_deriv_h = 0.001,
  verbose = FALSE
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()
ipw	ipw_output object from ipw.ate()
bandwidth_scale	Scaling of the calculated bandwidth, or in case of
kernel	Specifies which kernel function to be used
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
gauss_cutoff	Cutoff value for Gaussian kernel
num_deriv_h	Step size of numerical derivative.
verbose	Specifies if the program should print output while running.

Value

The variance of IPW

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Perform semiparametric inverse probability weighting
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 10,
           bwc_prop_score = 18)

# Calculate the variance of the Augmented IPW (AIPW)
var <- SDRcausal::ipw.var(x, y, trt, imp, ipw,
           bandwidth_scale = ipw$bw_pr)
```

nw_kernel_regress *The Nadaraya-Watson kernel estimator*

Description

Gives the expected value of Y given X = x by kernel regression according to the Nadaraya-Watson kernel estimator to get E(Y|X). Note that y and x may be vectors or matrices, as long as dim(x)[1] == dim(y)[1].

Usage

```
nw_kernel_regress(
  y,
  x,
  bandwidth = 1,
  kernel = "EPAN",
  gauss_cutoff = 0.001,
  verbose = FALSE
)
```

Arguments

y	Y in E(Y|X)
x	X in E(Y|X)
bandwidth	Kernel bandwidth
kernel	Indicates which kernel function to be used
gauss_cutoff	Cutoff value for Gaussian kernel
verbose	Specifies if the program should print output while running.

Value

Value of kernel regression

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes

# Extimating y given x, E(y | x)
k <- nw_kernel_regress(y, x, bandwidth = 1)
```

plot_imp *Plots imputation output*

Description

Plot function for visualisation of imputation output from imp.ate. Note: The function requires ggplot2.

Usage

```
plot_imp(x, y, treated, imp)
```

Arguments

x	Covariate matrix
y	Response vector
treated	Binary vetor indicating treatment
imp	imp_output object from imp.ate()

Value

A list of ggplot plots of observed and imputed values (pl_imp), imputed treated values vs CMS (pl_m1), and imputed untreated values vs CMS (pl_m0).

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
imp <- SDRcausal::imp.ate(x, y, trt, b1, b0,
           explicit_bandwidth = TRUE, bwc_dim_red1 = 1, bwc_impute1 = 1,
           bwc_dim_red0 = 1, bwc_impute0 = 1)

# Plotting
plots <- plot_imp(x, y, trt, imp)
```

plot_ipw *Plots IPW output*

Description

Plot function for visualisation of IPW output from ipw.ate. Note: The function requires ggplot2.

Usage

```
plot_ipw(treated, ipw)
```

Arguments

treated	Binary vetor indicating treatment
ipw	ipw_output object from ipw.ate()

Value

ggplot plot of the propensity score vs CMS.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
b1 <- SDRcausal::beta1_guess
b0 <- SDRcausal::beta0_guess
alp <- SDRcausal::alpha_guess

# Perform semiparametric imputation
ipw <- SDRcausal::ipw.ate(x, y, trt, alp, bwc_dim_red = 8,
          bwc_prop_score = 8)

# Plotting
plots <- plot_ipw(trt, ipw)
```

ps.semi *Estimates propensity score*

Description

Semiparametric estimation of the propensity score as in Ghosh, Ma, & De Luna (2020). To be used with SDRcausal::cms.ps.semi().

Usage

```
ps.semi(
  x,
  treated,
  alpha_hat,
  kernel = "EPAN",
  explicit_bandwidth = FALSE,
  bandwidth_scale = 1,
  verbose = FALSE
)
```

Arguments

x	Covariate matrix
treated	Binary vetor indicating treatment
alpha_hat	Locally efficient CMS
kernel	Kernel specification
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bw = bandwidth_scale * sd(x * beta) * n^(1/3).
bandwidth_scale	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the banddwidth.
verbose	Specifies if the program should print output while running.

Value

A list containing the estimated propensity scores values and their derivatives, and the bandwidth used.

References

Ghosh, T., Ma, Y., & De Luna, X. (2020). Sufficient dimension reduction for feasible and robust estimation of average causal effect. Statistica Sinica, accepted.

Examples

```
# Using example data from package SDRcausal
library(SDRcausal)

# Import example data
x <- SDRcausal::covariates
y <- SDRcausal::outcomes
trt <- SDRcausal::treated
alp <- SDRcausal::alpha_guess

# Perform semiparametric dimension reduction
cms <- SDRcausal::cms.ps.semi(x, trt, alp,
           explicit_bandwidth = TRUE, bandwidth_scale = 8)

# Estimate propensity score
pr_score <- SDRcausal::ps.semi(x, trt, cms$fa,
               bandwidth_scale = 8)
```

Index

∗Topic **data**
example_data, 10

aipw.ate, 2
aipw.var, 3
aipw2.ate, 4

b10_fun, 5
b_fun, 6

cms.ps.semi, 6
cms.semi, 8

example_data, 10

imp.ate, 10
imp.val, 13
imp.var, 14
imp2.var, 16
ipw.ate, 17
ipw.var, 19

nw_kernel_regress, 21

plot_imp, 22
plot_ipw, 23
ps.semi, 23