Package 'SDRcausal'

November 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` *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`	A binary vetor indicating treatment
`imp`	imp output object from `imp.ate`
`ipw`	ipw output object from `ipw.ate`

Value

The AIPW estimation (AIPW) of the average treatment effect (ATE).

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 = imp$bw1,
  bandwidth_scale0 = imp$bw0,
  bandwidth_scale_pr = ipw$bw_dr,
  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, or in case of explicit_bandwidth = TRUE, the actual bandwidth for the estimations of $E(\cdot\|\beta_1^T X)$. The default value is imp$bw1. If this default value is used, one should use the default value TRUE for explicit_bandwidth.
bandwidth_scale0	Scaling of the calculated bandwidth, or in case of explicit_bandwidth = TRUE, the actual bandwidth for the estimations of $E(\cdot\|\beta_0^T X)$. The default value is imp$bw0. If this default value is used, one should use the default value TRUE for explicit_bandwidth.
bandwidth_scale_pr	Scaling of the calculated bandwidth, or in case of explicit_bandwidth = TRUE, the actual bandwidth for the estimations of $E(\cdot\|\alpha^T X)$. The default value is ipw$bw_dr. If this default value is used, one should use the default value TRUE for explicit_bandwidth.
kernel	Specifies which kernel function to be used. The default is "EPAN".

`explicit_bandwidth`
: Specifies if `bandwidth_scale` will be used as the bandwidth or if it will be calculated as `bandwidth_scale` * sd($\beta^T x$) * $n^{(1/5)}$. The default value is `TRUE`.

`gauss_cutoff`
: Cutoff value for Gaussian kernel. The default value is `1e-3`.

`num_deriv_h`
: Step size of numerical derivative. The default value is `1e-6`.

`verbose`
: Specifies if the program should print output while running. The default value is `FALSE`.

Value

The variance of Augmented IPW estimator.

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 (AIPW2)*

Description

Combines IPW and IMP estimators to form the improved augmented IPW, AIPW2 as in Ghosh, Ma, & De Luna (2020).

Usage

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

Arguments

y	Observed response
treated	A binary vetor indicating treatment
imp	imp output object from imp.ate
ipw	ipw output object from ipw.ate

Value

The improved augmented IPW estimation (AIPW2) of the average treatment effect (ATE).

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)
aipw2 <- 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

covariates covariate matrix

outcomes observed outcome vector

treated binary treatment vector

beta1_guess Starting guess for CMS for treated

beta0_guess Starting guess for CMS for untreated

alpha_guess Starting 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 = TRUE,
  bwc_dim_red1 = 1,
  bwc_impute1 = 1.25,
  bwc_dim_red0 = 1,
  bwc_impute0 = 1.25,
  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`	A binary vector indicating treatment.
`beta_guess1`	Initial guess for β_1
`beta_guess0`	Initial guess for β_0
`solver`	Specifies which solver is to be used. Current options are `optim` and `cobyla` (from `nloptr` package). The diffault value is `"optim"`.
`kernel`	Specifies which kernel function is to be used, current options are: `"EPAN"`, `"QUARTIC"`, and `"GAUSSIAN"`. The default value is `"EPAN"`.
`explicit_bandwidth`	Specifies if `bandwidth_scale` will be used as the bandwidth or if it will be calculated as `bandwidth_scale` * sd($\beta^T x$) * $n^{(1/5)}$. The default value is `FALSE`.

`recalc_bandwidth`
: Specifies whether the bandwidth should be recalculated after the first stage (the estimations of dimension reduction step). If `explicit_bandwidth` is `TRUE`, `recalc_bandwidth` is not used, but if `explicit_bandwidth` is `FALSE`, then if `recalc_bandwidth` is `TRUE`, bandwidths are recalculated at the beginning of the second step based on `bwc_impute0` and `bwc_impute1`. If `recalc_bandwidth` is `FALSE`, the first step bandwidths are used. The default value is `FALSE`.

`bwc_dim_red1`
: Scaling of calculated bandwidth, or if `explicit_bandwidth = TRUE` used as the bandwidth. It is used in the dimension reduction step for $\hat{m}_1(\beta_1^T x)$. The default value is `1`.

`bwc_impute1`
: Scaling of calculated bandwidth, or if `explicit_bandwidth = TRUE` used as the bandwidth. It is used in the imputation step for $\hat{m}_1(\beta_1^T x)$. The default value is `1.25`.

`bwc_dim_red0`
: Scaling of calculated bandwidth, or if `explicit_bandwidth = TRUE` used as the bandwidth. It is used in the dimension reduction step for $\hat{m}_0(\beta_0^T x)$. The default value is `1`.

`bwc_impute0`
: Scaling of calculated bandwidth, or if `explicit_bandwidth = TRUE` used as the bandwidth. It is used in the imputation step for $\hat{m}_0(\beta_0^T x)$. The default value is `1.25`.

`gauss_cutoff`
: The cutoff value for Gaussian kernel. The default value is `1e-3`.

`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 times local linear regression fails. The default value is `10`.

`n_before_pen`
: The number of acceptable local linear regression failures during the estimation of β_0 and β_1 phase. The default value is `5`.

`to_extrapolate`
: Specifies whether to extrapolate or not. Since in $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$ estimates in terms of β_0 and β_1, local linear regression at the boundaries of $\beta_0^T x$ and $\beta_1^T x$ can be very volatile, it is recommended to use extrapolation on those points instead of local linear regression. The default value is `TRUE`.

`to_truncate`
: Specifies whether to truncate $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$ or not. After estimating $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$, if they are outside the range of observed outputs, they are replaced with the minimum and maximum observed outputs. The default value is `TRUE`.

`extrapolation_basis`
: The number of data points to base extrapolation on. Extrapolation at border points can be done based on a different number of neighborhood points. `extrapolation_basis` is how many neighborhood points are used. The default value is `5`.

`n_threads`
: Sets the number of threads for parallel computing. Set to 1 serial. If `n_threads` exceeds the maximum number of threads, sets `n_threads` to max_threads - 1. To use max_threads, set to `n_threads` to max_threads of system. The default value is `1`.

`verbose`
: Specifies if the program should print output while running. The default value is `TRUE`.

`...`
: Additional parameters passed to `optim` or `cobyla`.

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 = imp$bw1,
  bandwidth_scale0 = imp$bw0,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001
)
```

Arguments

`x`	Covariate matrix
`y`	Response vector
`treated`	A 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 of `explicit_bandwidth = TRUE`, the actual bandwidth for the estimation of $E(\cdot\|\beta_1^T X)$. The default value is `imp$bw1`. If this default value is used, one should use the default value `TRUE` for `explicit_bandwidth`.

`bandwidth_scale0`
: Scaling of the calculated bandwidth, or in case of `explicit_bandwidth = TRUE`, the actual bandwidth for the estimation of $E(\cdot|\beta_0^T X)$. The default value is `imp$bw0`. If this default value is used, one should use the default value `TRUE` for `explicit_bandwidth`.

`kernel`
: Specifies which kernel function to be used. The default is `"EPAN"`.

`explicit_bandwidth`
: Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as `bandwidth_scale` * sd($\beta_t^T x$) * $n^{(1/5)}$. The default value is `TRUE`.

`gauss_cutoff`
: The cutoff value for Gaussian kernel. The default value is `1e-3`.

Value

Variance of IMP estimator

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 IMP estimator.
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 = imp$bw1,
  bandwidth_scale0 = imp$bw0,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	A 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 of explicit_bandwidth = TRUE, the actual bandwidth for the estimation of $E(.\|\beta_1^T X)$. The default value is imp$bw1. If this default value is used, one should use the default value TRUE for explicit_bandwidth.
bandwidth_scale0	Scaling of the calculated bandwidth, or in case of explicit_bandwidth = TRUE, the actual bandwidth for the estimation of $E(.\|\beta_0^T X)$. The default value is imp$bw0. If this default value is used, one should use the default value TRUE for explicit_bandwidth.
kernel	Specifies which kernel function to be used. The default is "EPAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bandwidth_scale * sd($\beta_t^T x$) * $n^{(1/5)}$. The default value is TRUE.
gauss_cutoff	The cutoff value for Gaussian kernel. The default value is 1e-3.

Value

Variance of IMP2

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 IMP2 estimator.
var <- SDRcausal::imp2.var(x, y, trt, imp, ipw,
           bandwidth_scale1 = imp$bw1, bandwidth_scale0 = imp$bw0)
```

`inf.ate` *Performs Estimations of Average Treatment Effect and Infrences*

Description

Semiparametric estimation of the average treatment effect based on the all methods described in Ghosh, Ma, & De Luna (2020) and all infrences.

Usage

```
inf.ate(
  x,
  y,
  treated,
  beta_guess1,
  beta_guess0,
  imp.solver = "optim",
  imp.kernel = "EPAN",
  imp.explicit_bandwidth = FALSE,
  imp.recalc_bandwidth = TRUE,
  bwc_dim_red1 = 1,
  bwc_impute1 = 1.25,
  bwc_dim_red0 = 1,
  bwc_impute0 = 1.25,
  imp.gauss_cutoff = 0.001,
  imp.penalty = 10,
  imp.n_before_pen = 5,
  imp.to_extrapolate = TRUE,
  imp.to_truncate = TRUE,
  imp.extrapolation_basis = 5,
  alpha_initial,
```

```
  ipw.solver = "optim",
  ipw.kernel = "EPAN",
  ipw.explicit_bandwidth = FALSE,
  ipw.recalc_bandwidth = TRUE,
  bwc_dim_red = 1,
  bwc_prop_score = 10,
  ipw.gauss_cutoff = 0.001,
  ipw.penalty = 10,
  ipw.n_before_pen = 1,
  n_threads = 1,
  verbose = TRUE,
  imp.solver.options = NA,
  ipw.solver.options = NA
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	A binary vector indicating treatment status
beta_guess1	Initial guess for β_1
beta_guess0	Initial guess for β_0
imp.solver	Specifies which solver is to be used. Current options are optim and cobyla (from nloptr package). The diffault value is "optim".
imp.kernel	Specifies which kernel function is to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN". The default value is "EPAN".
imp.explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bandwidth_scale * sd($\beta^T x$) * $n^{(1/5)}$. The default value is FALSE.
imp.recalc_bandwidth	Specifies whether the bandwidth should be recalculated after the first stage (the estimations of dimension reduction step). If explicit_bandwidth is TRUE, recalc_bandwidth is not used, but if explicit_bandwidth is FALSE, then if recalc_bandwidth is TRUE, bandwidths are recalculated at the beginning of the second step based on bwc_impute0 and bwc_impute1. If recalc_bandwidth is FALSE, the first step bandwidths are used. The default value is FALSE.
bwc_dim_red1	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used in the dimension reduction step for $\hat{m}_1(\beta_1^T x)$. The default value is 1.
bwc_impute1	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used in the imputation step for $\hat{m}_1(\beta_1^T x)$. The default value is 1.25.
bwc_dim_red0	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used in the dimension reduction step for $\hat{m}_0(\beta_0^T x)$. The default value is 1.
bwc_impute0	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used in the imputation step for $\hat{m}_0(\beta_0^T x)$. The default value is 1.25.
imp.gauss_cutoff	The cutoff value for Gaussian kernel. The default value is 1e-3.

`imp.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 times local linear regression fails. The default value is `10`.

`imp.n_before_pen`
The number of acceptable local linear regression failures during the estimation of β_0 and β_1 phase. The default value is `5`.

`imp.to_extrapolate`
Specifies whether to extrapolate or not. Since in $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$ estimates in terms of β_0 and β_1, local linear regression at the boundaries of $\beta_0^T x$ and $\beta_1^T x$ can be very volatile, it is recommended to use extrapolation on those points instead of local linear regression. The default value is `TRUE`.

`imp.to_truncate`
Specifies whether to truncate $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$ or not. After estimating $\hat{m}_0(\beta_0^T x)$ and $\hat{m}_1(\beta_1^T x)$, if they are outside the range of observed outputs, they are replaced with the minimum and maximum observed outputs. The default value is `TRUE`.

`imp.extrapolation_basis`
The number of data points to base extrapolation on. Extrapolation at border points can be done based on a different number of neighborhood points. `extrapolation_basis` is how many neighborhood points are used. The default value is `5`.

`alpha_initial` Initial guess for α

`ipw.solver` Specifies which solver is to be used. Current options are `optim` and `cobyla` (from `nloptr` package). The diffault value is `"optim"`.

`ipw.kernel` Specifies which kernel function is to be used, current options are: `"EPAN"`, `"QUARTIC"`, and `"GAUSSIAN"`. The default value is `"EPAN"`.

`ipw.explicit_bandwidth`
Specifies if `bandwidth_scale` will be used as the bandwidth or if it will be calculated as `bandwidth_scale` * sd($\alpha^T x$) * $n^{(1/5)}$. The default value is `FALSE`.

`ipw.recalc_bandwidth`
Specifies whether the bandwidth should be recalculated after the estimations of α. If `explicit_bandwidth` is `TRUE`, `recalc_bandwidth` is not used, but if `explicit_bandwidth` is `FALSE`, then if `recalc_bandwidth` is `TRUE`, bandwidths are recalculated at the beginning of the second step based on `bwc_prop_score`. If `recalc_bandwidth` is `FALSE`, the first step bandwidths are used. The default value is `FALSE`.

`bwc_dim_red` Scaling of calculated bandwidth, or if `explicit_bandwidth` = `TRUE` used as the bandwidth. It is used in the dimension reduction step for $\alpha^T x$. The default value is `1`.

`bwc_prop_score` Scaling of calculated bandwidth, or if `explicit_bandwidth` = `TRUE` used as the bandwidth. It is used for the estimation of the propensity score. The default value is `10`.

`ipw.gauss_cutoff`
The cutoff value for Gaussian kernel. The default value is `1e-3`.

`ipw.penalty` Penalty for the optimizer if a probability is outside (0, 1) during the estimation of α phase. Added to the function value in solver as `penalty^(n - n_before_pen)`, where n is the number of probabilities outside (0, 1). The default value is `10`.

`ipw.n_before_pen`
The number of probabilities outside the range (0, 1) to accept during the estimation of α phase. The default value is `1`.

`n_threads` Sets the number of threads for parallel computing. Set to 1 serial. If `n_threads` exceeds the maximum number of threads, sets `n_threads` to max_threads - 1. To use max_threads, set to `n_threads` to max_threads of system. The default value is 1.

`verbose` Specifies if the program should print output while running. The default value is `TRUE`.

`imp.solver.options` Additional parameters passed to `optim` or `cobyla` for `imp.ate`.

`ipw.solver.options` Additional parameters passed to `optim` or `cobyla` for `ipw.ate`.

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
a <- SDRcausal::alpha_guess

# Perform semiparametric imputation
inf.ate <- SDRcausal::inf.ate(x, y, trt, b1, b0, alpha_initial = a)
```

`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,
  solver = "optim",
  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	A binary vector indicating treatment.
alpha_initial	Initial guess for α
solver	Specifies which solver is to be used. Current options are optim and cobyla (from nloptr package). The diffault value is "optim".
kernel	Specifies which kernel function is to be used, current options are: "EPAN", "QUARTIC", and "GAUSSIAN". The default value is "EPAN".
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as bandwidth_scale * sd($\alpha^T x$) * $n^{(1/5)}$. The default value is FALSE.
recalc_bandwidth	Specifies whether the bandwidth should be recalculated after the estimations of α. If explicit_bandwidth is TRUE, recalc_bandwidth is not used, but if explicit_bandwidth is FALSE, then if recalc_bandwidth is TRUE, bandwidths are recalculated at the beginning of the second step based on bwc_prop_score. If recalc_bandwidth is FALSE, the first step bandwidth is used. The default value is TRUE.
bwc_dim_red	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used in the dimension reduction step for $\alpha^T x$. The default value is 1.
bwc_prop_score	Scaling of calculated bandwidth, or if explicit_bandwidth = TRUE used as the bandwidth. It is used for the estimation of the propensity score. The default value is 10.
gauss_cutoff	The cutoff value for Gaussian kernel. The default value is 1e-3.
penalty	Penalty for the optimizer if a probability is outside (0, 1) during the estimation of α phase. Added to the function value in solver as penalty^(n - n_before_pen), where n is the number of probabilities outside (0, 1). The default value is 10.

n_before_pen	The number of probabilities outside the range (0, 1) to accept during the estimation of α phase. The default value is 1.
n_threads	Sets the number of threads for parallel computing. Set to 1 serial. If n_threads exceeds the maximum number of threads, sets n_threads to max_threads - 1. To use max_threads, set to n_threads to max_threads of system. The default value is 1.
verbose	Specifies if the program should print output while running. The default value is FALSE.
...	Additional parameters passed to optim or cobyla.

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)
```

---

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 = ipw$bw_dr,
  kernel = "EPAN",
  explicit_bandwidth = TRUE,
  gauss_cutoff = 0.001,
  num_deriv_h = 1e-06,
  verbose = FALSE
)
```

Arguments

x	Covariate matrix
y	Response vector
treated	A 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 `explicit_bandwidth = TRUE`, the actual bandwidth for the estimation of $E(\cdot\|\alpha^T X)$. The default value is `ipw$bw_dr`. If this default value is used, one should use the default value `TRUE` for `explicit_bandwidth`.
kernel	Specifies which kernel function to be used. The default is `"EPAN"`.
explicit_bandwidth	Specifies if bandwidth_scale will be used as the bandwidth or if it will be calculated as `bandwidth_scale` * sd($\alpha^T x$) * $n^{(1/5)}$. The default value is `TRUE`.
gauss_cutoff	The cutoff value for Gaussian kernel. The default value is `1e-3`.
num_deriv_h	Step size of numerical derivative. The default value is `1e-6`.
verbose	Specifies if the program should print output while running. The default value if `FALSE`.

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_dr)
```

`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

```
## S3 method for class 'imp'
plot(x, ..., covariates, y, treated)
```

Arguments

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

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
covariates <- 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(covariates, 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 , covariates = covariates, y=y, treated = trt)
```

`plot.ipw` *Plots IPW output*

Description

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

Usage

```
## S3 method for class 'ipw'
plot(x, ..., treated, covariates)
```

Arguments

`x`	ipw_output object from ipw.ate()
`...`	Other parameters
`treated`	Binary vetor indicating treatment
`covariates`	Covariate matrix

Value

ggplot plot of the propensity score vs CMS.

Examples

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

# Import example data
covariates <- 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(covariates, y, trt, alp, bwc_dim_red = 8,
          bwc_prop_score = 8)

# Plotting
plots <- plot(ipw, treated = trt, covariates = covariates)
```

`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