Package ‘IVDML’

December 12, 2025

Title Double Machine Learning with Instrumental Variables and Heterogeneity

Version 1.0.1

Description Instrumental variable (IV) estimators for homogeneous and heterogeneous treatment effects with efficient machine learning instruments. The estimators are based on double/debiased machine learning allowing for nonlinear and potentially high-dimensional control variables. Details can be found in Scheidegger, Guo and Bühlmann (2025) ``Inference for heterogeneous treatment effects with efficient instruments and machine learning'' <doi:10.48550/arXiv.2503.03530>.

URL `https://github.com/cyrillsch/IVDML`

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports mgcv, ranger, stats, xgboost (>= 3.1.2.1)

NeedsCompilation no

Author Cyrill Scheidegger [aut, cre, cph] (ORCID: <`https://orcid.org/0009-0005-2851-1384`>)

Maintainer Cyrill Scheidegger <`cyrill.scheidegger@stat.math.ethz.ch`>

Repository CRAN

Date/Publication 2025-12-12 10:50:06 UTC

Contents

`bandwidth_normal` *Compute Bandwidth Using the Normal Reference Rule*

Description

This function calculates the bandwidth for kernel smoothing using the Normal Reference Rule. The rule is based on Silverman's rule of thumb, which selects the bandwidth as a function of the standard deviation and interquartile range (IQR) of the data. The bandwidth is computed as: $h = 1.06 \times \min(\mathrm{sd}(A), \mathrm{IQR}(A)/1.34)/N^{0.2}$, where $\mathrm{sd}(A)$ is the standard deviation of `A`, $\mathrm{IQR}(A)$ is the interquartile range and `N` is the length of `A`.

Usage

```
bandwidth_normal(A)
```

Arguments

`A`	Numeric vector. The data for which the bandwidth is to be computed.

Value

A numeric value representing the computed bandwidth.

References

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC monographs on statistics and applied probability. Chapman & Hall.

Examples

```
set.seed(1)
A <- rnorm(100)
bandwidth_normal(A)
```

`coef.IVDML` *Extract Treatment Effect Estimate from an IVDML Object*

Description

This function computes the estimated (potentially heterogeneous) treatment effect from a fitted `IVDML` object (output of `fit_IVDML()`).

Usage

```
## S3 method for class 'IVDML'
coef(
  object,
  iv_method,
  a = NULL,
  A = NULL,
  kernel_name = NULL,
  bandwidth = NULL,
  ...
)
```

Arguments

`object`	An object of class `IVDML`, produced by the `fit_IVDML()` function.
`iv_method`	Character. The instrumental variable estimation method to use. Must be one of the methods specified in the fitted object.
`a`	Numeric (optional). A specific value of `A` at which to evaluate the heterogeneous treatment effect. If `NULL`, the function returns the homogeneous treatment effect.
`A`	Numeric vector (optional). The variable with respect to which treatment effect heterogeneity is considered. If `NULL`, the function assumes the `A` used in model fitting.
`kernel_name`	Character (optional). The name of the kernel function to use for smoothing (if a heterogeneous treatment effect is estimated). Needs to be one of "boxcar", "gaussian", "epanechnikov" or "tricube".
`bandwidth`	Numeric (optional). The bandwidth for the kernel smoothing (if a heterogeneous treatment effect is estimated).
`...`	Further arguments passed to or from other methods.

Value

If `a` is not specified, the estimated homogeneous treatment effect is returned. If `a` is specified, the heterogeneous treatment effect $\beta(a)$ at $A = a$ is returned.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, ml_method = "gam")
coef(fit, iv_method = "mlIV")
coef(fit, iv_method = "mlIV", a = 0, A = A, kernel_name = "boxcar", bandwidth = 0.2)
```

`fit_IVDML` *Fitting Double Machine Learning Models with Instrumental Variables and Potentially Heterogeneous Treatment Effect*

Description

This function is used to fit a Double Machine Learning (DML) model with Instrumental Variables (IV) with the goal to perform inference on potentially heterogeneous treatment effects. The model under study is $Y = \beta(A)D + g(X) + \epsilon$, where the error ϵ is potentially correlated with the treatment D, but there is an IV Z satisfying $\mathbb{E}[\epsilon|Z, X] = 0$. The object of interest is the treatment effect β of the treatment D on the response Y. The treatment effect β is either constant or can depend on the univariate quantity A, which is typically a component of the covariates X.

Usage

```
fit_IVDML(
  Y,
  D,
  Z,
  X = NULL,
  A = NULL,
  ml_method,
  ml_par = list(),
  A_deterministic_X = TRUE,
  K_dml = 5,
  iv_method = c("linearIV", "mlIV"),
  S_split = 1
)
```

Arguments

`Y`	Numeric vector. Response variable.
`D`	Numeric vector. Treatment variable.
`Z`	Matrix, vector, or data frame. Instrumental variables.

`X`	Matrix, vector, or data frame. Additional covariates (default: NULL).
`A`	Numeric vector. Variable with respect to which treatment effect heterogeneity is considered. Usually equal to a column of X and in this case it can also be specified later (default: NULL).
`ml_method`	Character. Machine learning method to use. Options are "gam", "xgboost", and "randomForest".
`ml_par`	List. Parameters for the machine learning method: • If `ml_method == "gam"`, can specify `ind_lin_Z` and `ind_lin_X` for components of `Z` and `X` to be modeled linearly. • If `ml_method == "xgboost"`, can specify `max_nrounds`, `k_cv`, `early_stopping_rounds`, and vectors `eta` and `max_depth`. • If `ml_method == "randomForest"`, can specify `num.trees`, `num_mtry` (number of different mtry values to try out) or a vector `mtry`, a vector `max.depth`, `num_min.node.size` (number of different min.node.size values to try out) or a vector `min.node.size`. • To specify different parameters for the different nuisance function regressions, `ml_par` should be a list of lists: `ml_par_D_XZ` (parameters for nuisance function $\mathbb{E}[D\vert Z,X]$, needed for `iv_method` "mlIV" and "mlIV_direct"), `ml_par_D_X` (parameters for nuisance function $\mathbb{E}[D\vert X]$, needed for `iv_method` "linearIV", "mlIV" and "mlIV_direct"), `ml_par_f_X` (parameters for nuisance function $\mathbb{E}[\widehat{\mathbb{E}}[D\vert Z,X]\vert X]$, needed for `iv_method` "mlIV"), `ml_par_Y_X` (parameters for nuisance function $\mathbb{E}[Y\vert X]$, needed for `iv_method` "linearIV", "mlIV" and "mlIV_direct"), `ml_par_Z_X` (parameters for nuisance function $\mathbb{E}[Z\vert X]$, needed for `iv_method` "linearIV").
`A_deterministic_X`	Logical. Whether `A` is a deterministic function of `X` (default: TRUE).
`K_dml`	Integer. Number of cross-fitting folds (default: 5).
`iv_method`	Character vector. Instrumental variables estimation method. Options: "linearIV", "mlIV", "mlIV_direct" (default: c("linearIV", "mlIV")). "linearIV" corresponds to using instruments linearly and "mlIV" corresponds to using machine learning instruments. "mlIV_direct" is a variant of "mlIV" that uses the same estimate of $\mathbb{E}[D\vert X]$ for both the residuals $X - \mathbb{E}[D\vert X]$ and $\mathbb{E}[D\vert Z,X] - \mathbb{E}[D\vert X]$, whereas "mlIV" uses a two-stage estimate of $\mathbb{E}[\widehat{\mathbb{E}}[D\vert Z,X]\vert X]$ for the residuals $\mathbb{E}[D\vert Z,X] - \mathbb{E}[D\vert X]$.
`S_split`	Integer. Number of sample splits for cross-fitting (default: 1).

Value

An object of class `IVDML`, containing:

- `results_splits`: A list of S_split lists of cross-fitted residuals from the different sample splits.
- `A`: The argument `A` of the function.
- `ml_method`: The argument `ml_method` of the function.
- `A_deterministic_X`: The argument `A_deterministic_X` of the function.

- `iv_method`: The argument `iv_method` of the function. The treatment effect estimates, standard errors and confidence intervals can be calculated from the `IVDML` object using the functions `coef.IVDML()`, `se()`, `standard_confint()`, `robust_confint()`.

References

Cyrill Scheidegger, Zijian Guo and Peter Bühlmann. Inference for heterogeneous treatment effects with efficient instruments and machine learning. Preprint, arXiv:2503.03530, 2025.

See Also

Inference for a fitted `IVDML` object is done with the functions `coef.IVDML()`, `se()`, `standard_confint()` and `robust_confint()`.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, A = A, ml_method = "gam")
coef(fit, iv_method = "mlIV", a = 0, A = A, kernel_name = "boxcar", bandwidth = 0.2)
```

`print.IVDML` *Print IVDML*

Description

Print information for an IVDML object.

Usage

```
## S3 method for class 'IVDML'
print(x, ...)
```

Arguments

`x`	Fitted object of class `IVDML`.
`...`	Further arguments passed to or from other methods.

Value

No return value, called for side effects

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, A = A, ml_method = "gam")
print(fit)
```

`robust_confint` *Compute Robust Confidence Interval for Treatment Effect in an IVDML Object*

Description

This function computes a robust (with respect to weak IV) confidence interval/confidence set for the estimated treatment effect in a fitted `IVDML` object (output of `fit_IVDML()`). The confidence interval/confidence set is constructed by inverting the robust test from the `robust_p_value_aggregated()` function, which either uses the Double Machine Learning aggregation method (`"DML_agg"`) or the quantile-based method of Meinshausen, Meier, and Bühlmann (2009) (`"MMB_agg"`) to aggregate the p-values corresponding to the `S_split` cross-fitting sample splits (where `S_split` was an argument of the `fit_IVDML()` function).

Usage

```
robust_confint(
  object,
  iv_method,
  level = 0.95,
  a = NULL,
  A = NULL,
  kernel_name = NULL,
  bandwidth = NULL,
  CI_range = NULL,
  agg_method = "DML_agg",
  gamma = 0.5
)
```

Arguments

`object`	An object of class `IVDML`, produced by the `fit_IVDML()` function.
`iv_method`	Character. The instrumental variable estimation method to use. Must be one of the methods specified in the fitted object.
`level`	Numeric (default: 0.95). The confidence level for the confidence interval.

`a`	Numeric (optional). A specific value of `A` at which to compute the confidence interval for the heterogeneous treatment effect. If `NULL`, the function returns the confidence interval for the homogeneous treatment effect.
`A`	Numeric vector (optional). The variable with respect to which treatment effect heterogeneity is considered. If `NULL`, the function assumes the `A` used in model fitting.
`kernel_name`	Character (optional). The name of the kernel function to use for smoothing (if a heterogeneous treatment effect is estimated). Must be one of `"boxcar"`, `"gaussian"`, `"epanechnikov"`, or `"tricube"`.
`bandwidth`	Numeric (optional). The bandwidth for the kernel smoothing (if a heterogeneous treatment effect is estimated).
`CI_range`	Numeric vector of length 2 (optional). The search range for the confidence interval. If `NULL`, the function sets `CI_range` to be four times as large as the standard confidence interval centered at the point estimate of the treatment effect.
`agg_method`	Character (default: `"DML_agg"`). The aggregation method for computing the confidence interval. Options are: • `"DML_agg"`: Uses the Double Machine Learning (DML) aggregation approach. • `"MMB_agg"`: Uses the quantile-based aggregation method of Meinshausen, Meier, and Bühlmann (2009).
`gamma`	Numeric (default: 0.5). Quantile level for the `"MMB_agg"` method. Ignored if `agg_method = "DML_agg"`.

Value

A list with the following elements:

- `CI`: A named numeric vector with the lower and upper bounds of the confidence interval.
- `level`: The confidence level used.
- `message`: A message describing the nature of the confidence set (e.g., whether it spans the full range, is non-connected, or is empty due to optimization failure).
- `heterogeneous_parameters`: A list of parameters (`a`, `kernel_name`, `bandwidth`) if a heterogeneous treatment effect is considered; otherwise, `NULL`.

References

Meinshausen, N., Meier, L., & Bühlmann, P. (2009). *P-values for high-dimensional regression*. Journal of the American Statistical Association, 104(488), 1671–1681.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
```

```
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, ml_method = "gam")
robust_confint(fit, iv_method = "mlIV", CI_range = c(-10, 10))
robust_confint(fit, iv_method = "mlIV", a = 0, A = A,
               kernel_name = "boxcar", bandwidth = 0.2, CI_range = c(-10, 10))
```

`robust_p_value_aggregated` *Compute Aggregated Robust p-Value for Treatment Effect in an IVDML Object*

Description

This function calculates an aggregated robust (with respect to weak IV) p-value for testing a candidate treatment effect value in a fitted `IVDML` object (output of `fit_IVDML()`), using either the the standard Double Machine Learning aggregation method ("DML_agg") or the method by Meinshausen, Meier, and Bühlmann (2009) ("MMB_agg") to aggregate the p-values corresponding to the `S_split` cross-fitting sample splits (where `S_split` was an argument of the `fit_IVDML()` function).

Usage

```
robust_p_value_aggregated(
  object,
  candidate_value,
  iv_method,
  a = NULL,
  A = NULL,
  kernel_name = NULL,
  bandwidth = NULL,
  agg_method = "DML_agg",
  gamma = 0.5
)
```

Arguments

`object`	An object of class `IVDML`, produced by the `fit_IVDML()` function.
`candidate_value`	Numeric. The candidate treatment effect value to test.
`iv_method`	Character. The instrumental variable estimation method to use. Must be one of the methods specified in the fitted object.
`a`	Numeric (optional). A specific value of `A` at which to compute the p-value for the heterogeneous treatment effect. If `NULL`, the function returns the p-value for the homogeneous treatment effect.

`A`	Numeric vector (optional). The variable with respect to which treatment effect heterogeneity is considered. If `NULL`, the function assumes the `A` used in model fitting.
`kernel_name`	Character (optional). The name of the kernel function to use for smoothing (if a heterogeneous treatment effect is estimated). Must be one of "boxcar", "gaussian", "epanechnikov", or "tricube".
`bandwidth`	Numeric (optional). The bandwidth for the kernel smoothing (if a heterogeneous treatment effect is estimated).
`agg_method`	Character (default: "DML_agg"). The aggregation method for computing the p-value. Options are: • `"DML_agg"`: Uses the Double Machine Learning (DML) aggregation approach. • `"MMB_agg"`: Uses the quantile-based aggregation method of Meinshausen, Meier, and Bühlmann (2009).
`gamma`	Numeric (default: 0.5). Quantile level for the `"MMB_agg"` method. Ignored if `agg_method = "DML_agg"`.

Value

The aggregated robust p-value for testing the candidate treatment effect.

References

Meinshausen, N., Meier, L., & Bühlmann, P. (2009). *P-values for high-dimensional regression*. Journal of the American Statistical Association, 104(488), 1671–1681.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, A = A, ml_method = "gam")
robust_p_value_aggregated(fit, candidate_value = 0, iv_method = "mlIV")
robust_p_value_aggregated(fit, candidate_value = 0, iv_method = "mlIV",
                          a = 0, A = A, kernel_name = "boxcar", bandwidth = 0.2)
```

`se` *Compute Standard Error for the Treatment Effect Estimate in an IVDML Object*

Description

This function calculates the standard error of the estimated (potentially heterogeneous) treatment effect from a fitted `IVDML` object (output of `fit_IVDML()`).

Usage

```
se(object, iv_method, a = NULL, A = NULL, kernel_name = NULL, bandwidth = NULL)
```

Arguments

`object`	An object of class `IVDML`, produced by the `fit_IVDML()` function.
`iv_method`	Character. The instrumental variable estimation method to use. Must be one of the methods specified in the fitted object.
`a`	Numeric (optional). A specific value of `A` at which to evaluate the standard error of the heterogeneous treatment effect. If `NULL`, the function returns the standard error of the homogeneous treatment effect.
`A`	Numeric vector (optional). The variable with respect to which treatment effect heterogeneity is considered. If `NULL`, the function assumes the `A` used in model fitting.
`kernel_name`	Character (optional). The name of the kernel function to use for smoothing (if a heterogeneous treatment effect is estimated). Must be one of "boxcar", "gaussian", "epanechnikov", or "tricube".
`bandwidth`	Numeric (optional). The bandwidth for the kernel smoothing (if a heterogeneous treatment effect is estimated).

Value

A numeric value representing the estimated standard error of the treatment effect estimate. If `a` is not specified, the function returns the standard error of the homogeneous treatment effect. If `a` is specified, it returns the standard error of the heterogeneous treatment effect estimate at $A = a$.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, ml_method = "gam")
se(fit, iv_method = "mlIV")
se(fit, iv_method = "mlIV", a = 0, A = A, kernel_name = "boxcar", bandwidth = 0.2)
```

`standard_confint` *Compute Standard Confidence Interval for the Treatment Effect Estimate in an IVDML Object*

Description

This function calculates a standard confidence interval for the estimated (potentially heterogeneous) treatment effect from a fitted `IVDML` object (output of `fit_IVDML()`). The confidence interval is computed using the normal approximation method using the standard error computed by `se()` and the treatment effect estimate from `coef()`.

Usage

```
standard_confint(
  object,
  iv_method,
  a = NULL,
  A = NULL,
  kernel_name = NULL,
  bandwidth = NULL,
  level = 0.95
)
```

Arguments

`object`	An object of class `IVDML`, produced by the `fit_IVDML()` function.
`iv_method`	Character. The instrumental variable estimation method to use. Must be one of the methods specified in the fitted object.
`a`	Numeric (optional). A specific value of `A` at which to compute the confidence interval for the heterogeneous treatment effect. If `NULL`, the function returns the confidence interval for the homogeneous treatment effect.
`A`	Numeric vector (optional). The variable with respect to which treatment effect heterogeneity is considered. If `NULL`, the function assumes the `A` used in object fitting.
`kernel_name`	Character (optional). The name of the kernel function to use for smoothing (if a heterogeneous treatment effect is estimated). Must be one of "boxcar", "gaussian", "epanechnikov", or "tricube".
`bandwidth`	Numeric (optional). The bandwidth for the kernel smoothing (if a heterogeneous treatment effect is estimated).
`level`	Numeric (default: 0.95). The confidence level for the interval (e.g., 0.95 for a 95% confidence interval).

Value

description A list containing:

- `CI`: A numeric vector of length 2 with the lower and upper confidence interval bounds.
- `level`: The confidence level used.
- `heterogeneous_parameters`: A list with values of `a`, `kernel_name`, and `bandwidth` (if applicable), or `NULL` if a homogeneous treatment effect is estimated.

Examples

```
set.seed(1)
Z <- rnorm(100)
X <- Z + rnorm(100)
H <- rnorm(100)
D <- Z^2 + sin(X) + H + rnorm(100)
A <- X
Y <- tanh(A) * D + cos(X) - H + rnorm(100)
fit <- fit_IVDML(Y = Y, D = D, Z = Z, X = X, ml_method = "gam")
standard_confint(fit, iv_method = "mlIV")
standard_confint(fit, iv_method = "mlIV", a = 0, A = A,
                 kernel_name = "boxcar", bandwidth = 0.2, level = 0.95)
```

Index