Package 'MLBC'

June 4, 2025

Version 0.2.1

Title Bias Correction Methods for Models Using Synthetic Data

Description Implements three bias-correction techniques from Battaglia et al. (2025 <doi:10.48550/arXiv.2402.15585>) to improve inference in regression models with covariates generated by AI or machine learning.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown=TRUE)

RoxygenNote 7.3.2

Imports TMB, MASS, numDeriv, stats

LinkingTo TMB, RcppEigen

Suggests roxygen2

Depends R (>= 3.5)

LazyData true

Contents

`ols` *Ordinary Least Squares (OLS) regression*

Description

Ordinary Least Squares regression with support for both formula and array-based interfaces. This function provides a unified interface for fitting linear models using either R formulas with data frames or raw matrices.

Usage

```
ols(Y, X = NULL, data = parent.frame(), se = TRUE, intercept = FALSE, ...)

## Default S3 method:
ols(Y, X, data = parent.frame(), se = TRUE, intercept = FALSE, ...)

## S3 method for class 'formula'
ols(Y, X = NULL, data = parent.frame(), se = TRUE, intercept = TRUE, ...)
```

Arguments

`Y`	numeric response vector, or a one-sided formula
`X`	numeric design matrix (if `Y` is numeric)
`data`	data frame (if `Y` is a formula)
`se`	logical; return heteroskedastic-robust standard errors?
`intercept`	logical; include an intercept term?
`...`	unused

Value

An object of class `mlbc_fit` and `mlbc_ols` with:

- `coef`: coefficient estimates
- `vcov`: variance-covariance matrix
- `sXX`: scaled cross-product X'X / n

Usage Options

Option 1: Formula Interface

- `Y`: A one-sided formula (e.g., `y ~ x1 + x2`)
- `data`: A data frame containing the variables referenced in the formula

Option 2: Array Interface

- `Y`: Response variable vector
- `X`: Design matrix of covariates

Examples

```
# Load the remote work dataset
data(SD_data)

# Formula interface
fit1 <- ols(log(salary) ~ wfh_wham + soc_2021_2 + employment_type_name,
            data = SD_data)
summary(fit1)

# Array interface
Y <- log(SD_data$salary)
X <- model.matrix(~ wfh_wham + soc_2021_2, data = SD_data)
fit2 <- ols(Y, X[, -1], intercept = TRUE)  # exclude intercept column
summary(fit2)
```

`ols_bca` *Additive bias-corrected OLS (BCA)*

Description

Performs an additive bias correction to regressions that include a binary covariate generated by AI/ML. This method requires an external estimate of the false-positive rate. Standard errors are adjusted to account for uncertainty in the false-positive rate estimate.

Usage

```
ols_bca(
  Y,
  Xhat = NULL,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)

## Default S3 method:
ols_bca(
  Y,
  Xhat,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)

## S3 method for class 'formula'
ols_bca(
```

```
  Y,
  Xhat = NULL,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)
```

Arguments

`Y`	numeric response vector, or a one-sided formula
`Xhat`	numeric matrix of regressors (if `Y` is numeric); the ML-regressor is column `gen_idx`
`fpr`	numeric; estimated false-positive rate of the ML regressor
`m`	integer; size of the external sample used to estimate the classifier's false-positive rate. Can be set to a large number when the false-positive rate is known exactly
`data`	data frame (if `Y` is a formula)
`intercept`	logical; if `TRUE`, prepends a column of 1's to `Xhat`
`gen_idx`	integer; 1-based index of the ML-generated variable to apply bias correction to. If not specified, defaults to the first non-intercept variable
`...`	unused

Value

An object of class `mlbc_fit` and `mlbc_bca` with:

- `coef`: bias-corrected coefficient estimates (ML-slope first, other slopes, intercept last)
- `vcov`: adjusted variance-covariance matrix for those coefficients

Usage Options

Option 1: Formula Interface

- `Y`: A one-sided formula string
- `data`: Data frame containing the variables referenced in the formula

Option 2: Array Interface

- `Y`: Response variable vector
- `Xhat`: Design matrix of covariates

Examples

```
# Load the remote work dataset
data(SD_data)

# Formula interface
fit_bca <- ols_bca(log(salary) ~ wfh_wham + soc_2021_2 + employment_type_name,
                   data = SD_data,
                   fpr = 0.009,  # estimated false positive rate
```

```
                    m = 1000)     # validation sample size
summary(fit_bca)

# Array interface
Y <- log(SD_data$salary)
Xhat <- model.matrix(~ wfh_wham + soc_2021_2, data = SD_data)[, -1]
fit_bca2 <- ols_bca(Y, Xhat, fpr = 0.009, m = 1000, intercept = TRUE)
summary(fit_bca2)
```

`ols_bca_topic` *Additive bias-corrected OLS for topic models (BCA-Topic)*

Description

Bias-corrected additive estimator for topic model regression. This method applies additive bias correction to regressions that include topic proportions as covariates, accounting for estimation uncertainty in the topic model.

Usage

```
ols_bca_topic(
  Y,
  Q = NULL,
  W,
  S,
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)

## Default S3 method:
ols_bca_topic(
  Y,
  Q = NULL,
  W,
  S,
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)

## S3 method for class 'formula'
ols_bca_topic(
  Y,
  Q = NULL,
  W,
  S,
```

```
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)
```

Arguments

Y	numeric response vector, or a one-sided formula
Q	numeric matrix of additional controls (if Y is numeric)
W	numeric matrix of document-term frequencies
S	numeric matrix of topic loadings
B	numeric matrix of topic-word distributions
k	numeric; bias correction parameter
data	data frame (if Y is a formula)
intercept	logical; if TRUE, includes an intercept term
...	additional arguments

Value

An object of class `mlbc_fit` and `mlbc_bca_topic` with:

- `coef`: bias-corrected coefficient estimates
- `vcov`: adjusted variance-covariance matrix

Examples

```
# Load topic model dataset
data(topic_model_data)

# Extract components
Y <- topic_model_data$estimation_data$ly
Z <- as.matrix(topic_model_data$covars)
theta_full <- as.matrix(topic_model_data$theta_est_full)
beta_full <- as.matrix(topic_model_data$beta_est_full)
lda_data <- as.matrix(topic_model_data$lda_data)

# Apply additive bias correction
kappa <- mean(1.0 / lda_data[, 1]) * sqrt(nrow(lda_data))
S <- matrix(c(1.0, 0.0), nrow = 1)

fit <- ols_bca_topic(Y, Z, theta_full, S, beta_full, k = kappa)
summary(fit)
```

`ols_bcm` *Multiplicative bias-corrected OLS (BCM)*

Description

Performs a multiplicative bias correction to regressions that include a binary covariate generated by AI/ML. This method requires an external estimate of the false-positive rate. Standard errors are adjusted to account for uncertainty in the false-positive rate estimate.

Usage

```
ols_bcm(
  Y,
  Xhat = NULL,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)

## Default S3 method:
ols_bcm(
  Y,
  Xhat,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)

## S3 method for class 'formula'
ols_bcm(
  Y,
  Xhat = NULL,
  fpr,
  m,
  data = parent.frame(),
  intercept = TRUE,
  gen_idx = 1,
  ...
)
```

Arguments

`Y`	numeric response vector, or a one-sided formula
`Xhat`	numeric matrix of regressors (if `Y` is numeric); the ML-regressor is column `gen_idx`

fpr	numeric; estimated false-positive rate of the ML regressor
m	integer; size of the external sample used to estimate the classifier's false-positive rate. Can be set to a large number when the false-positive rate is known exactly
data	data frame (if Y is a formula)
intercept	logical; if TRUE, prepends a column of 1's to Xhat
gen_idx	integer; 1-based index of the ML-generated variable to apply bias correction to. If not specified, defaults to the first non-intercept variable
...	unused

Value

An object of class mlbc_fit and mlbc_bcm with:

- coef: bias-corrected coefficient estimates (ML-slope first, other slopes, intercept last)
- vcov: adjusted variance-covariance matrix for those coefficients

Usage Options

Option 1: Formula Interface

- Y: A one-sided formula string
- data: Data frame containing the variables referenced in the formula

Option 2: Array Interface

- Y: Response variable vector
- Xhat: Design matrix of covariates

Examples

```
# Load the remote work dataset
data(SD_data)

# Formula interface
fit_bcm <- ols_bcm(log(salary) ~ wfh_wham + soc_2021_2 + employment_type_name,
                   data = SD_data,
                   fpr = 0.009,  # estimated false positive rate
                   m = 1000)     # validation sample size
summary(fit_bcm)

# Compare with uncorrected OLS
fit_ols <- ols(log(salary) ~ wfh_wham + soc_2021_2 + employment_type_name,
               data = SD_data)

# Display coefficient comparison
data.frame(
  OLS = coef(fit_ols)[1:2],
  BCM = coef(fit_bcm)[1:2]
)
```

`ols_bcm_topic` *Multiplicative bias-corrected OLS for topic models (BCM-Topic)*

Description

Bias-corrected multiplicative estimator for topic model regression. This method applies multiplicative bias correction to regressions that include topic proportions as covariates, accounting for estimation uncertainty in the topic model.

Usage

```
ols_bcm_topic(
  Y,
  Q = NULL,
  W,
  S,
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)

## Default S3 method:
ols_bcm_topic(
  Y,
  Q = NULL,
  W,
  S,
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)

## S3 method for class 'formula'
ols_bcm_topic(
  Y,
  Q = NULL,
  W,
  S,
  B,
  k,
  data = parent.frame(),
  intercept = TRUE,
  ...
)
```

Arguments

`Y`	numeric response vector, or a one-sided formula

Q	numeric matrix of additional controls (if Y is numeric)
W	numeric matrix of document-term frequencies
S	numeric matrix of topic loadings
B	numeric matrix of topic-word distributions
k	numeric; bias correction parameter
data	data frame (if Y is a formula)
intercept	logical; if TRUE, includes an intercept term
...	additional arguments

Value

An object of class `mlbc_fit` and `mlbc_bcm_topic` with:

- `coef`: bias-corrected coefficient estimates
- `vcov`: adjusted variance-covariance matrix

Examples

```
# Load topic model dataset
data(topic_model_data)

# Extract components
Y <- topic_model_data$estimation_data$ly
Z <- as.matrix(topic_model_data$covars)
theta_full <- as.matrix(topic_model_data$theta_est_full)
beta_full <- as.matrix(topic_model_data$beta_est_full)
lda_data <- as.matrix(topic_model_data$lda_data)

# Apply multiplicative bias correction
kappa <- mean(1.0 / lda_data[, 1]) * sqrt(nrow(lda_data))
S <- matrix(c(1.0, 0.0), nrow = 1)

fit <- ols_bcm_topic(Y, Z, theta_full, S, beta_full, k = kappa)
summary(fit)
```

---

one_step *One-step maximum likelihood estimation*

Description

Maximum likelihood estimation of the regression model, treating the generated covariate as a noisy proxy for the true latent variable. This method is particularly useful when an estimate of the false positive rate is not available. The variance of the estimates is approximated via the inverse Hessian at the optimum.

Usage

```
one_step(
  Y,
  Xhat = NULL,
  homoskedastic = FALSE,
  distribution = c("normal", "t", "laplace", "gamma", "beta"),
  nu = 4,
  gshape = 2,
  gscale = 1,
  ba = 2,
  bb = 2,
  intercept = TRUE,
  gen_idx = 1,
  data = parent.frame(),
  ...
)

## Default S3 method:
one_step(
  Y,
  Xhat,
  homoskedastic = FALSE,
  distribution = c("normal", "t", "laplace", "gamma", "beta"),
  nu = 4,
  gshape = 2,
  gscale = 1,
  ba = 2,
  bb = 2,
  intercept = TRUE,
  gen_idx = 1,
  ...
)

## S3 method for class 'formula'
one_step(
  Y,
  Xhat = NULL,
  homoskedastic = FALSE,
  distribution = c("normal", "t", "laplace", "gamma", "beta"),
  nu = 4,
  gshape = 2,
  gscale = 1,
  ba = 2,
  bb = 2,
  intercept = TRUE,
  gen_idx = 1,
  data = parent.frame(),
  ...
)
```

Arguments

Y	numeric response vector, or a one-sided formula
Xhat	numeric matrix of regressors (if Y is numeric)
homoskedastic	logical; if TRUE, assumes a common error variance; otherwise, the error variance is allowed to vary with the true latent binary variable
distribution	character; distribution for error terms. One of "normal", "t", "laplace", "gamma", "beta"
nu	numeric; degrees of freedom (for Student-t distribution)
gshape	numeric; shape parameter (for Gamma distribution)
gscale	numeric; scale parameter (for Gamma distribution)
ba	numeric; alpha parameter (for Beta distribution)
bb	numeric; beta parameter (for Beta distribution)
intercept	logical; if TRUE, prepend an intercept column to Xhat
gen_idx	integer; index (1-based) of the binary ML-generated variable. If not specified, defaults to the first non-intercept variable
data	data frame (if Y is a formula)
...	unused

Value

An object of class mlbc_fit and mlbc_onestep with:

- coef: estimated regression coefficients
- vcov: variance-covariance matrix

Usage Options

Option 1: Formula Interface

- Y: A one-sided formula string
- data: Data frame containing the variables referenced in the formula

Option 2: Array Interface

- Y: Response variable vector
- Xhat: Design matrix of covariates

Examples

```
# Load the remote work dataset
data(SD_data)

# Basic one-step estimation
fit_onestep <- one_step(log(salary) ~ wfh_wham + soc_2021_2 + employment_type_name,
                        data = SD_data)
summary(fit_onestep)

# With different error distribution
fit_t <- one_step(log(salary) ~ wfh_wham + soc_2021_2,
                  data = SD_data,
```

```
                  distribution = "t",
                  nu = 4)
summary(fit_t)

# Homoskedastic errors
fit_homo <- one_step(log(salary) ~ wfh_wham + soc_2021_2,
                     data = SD_data,
                     homoskedastic = TRUE)
summary(fit_homo)
```

SD_data *Job postings dataset*

Description

A subset of data relating to job postings on the Lightcast platform for demonstrating bias correction methods with ML-generated variables.

Usage

```
SD_data
```

Format

SD_data**:**

A data frame with 16315 rows and 7 columns:

city_name Character. City of the job posting

naics_2022_2 Character. Type of business (NAICS industry classification)

id Integer. Unique identifier of the job posting

salary Numeric. Salary offered (response variable)

wfh_wham Numeric. Binary label generated via ML, indicating whether remote work is offered (subject to measurement error)

soc_2021_2 Character. Occupation code (SOC classification)

employment_type_name Character. Employment type (part time/full time)

Source

Proprietary data from Lightcast job postings platform

Examples

```
## Not run:
data(SD_data)
fit <- ols_bca(log(salary) ~ wfh_wham + soc_2021_2 + naics_2022_2,
               data = SD_data, fpr = 0.009, m = 1000)

## End(Not run)
```

`topic_model_data` *Topic model dataset*

Description

Dataset containing topic model outputs for demonstrating bias correction methods in topic model regressions using CEO diary data.

Usage

```
topic_model_data
```

Format

A list with 8 components:

covars Data frame (916 x 11): Control variables

estimation_data Data frame (916 x 672): Contains outcome `ly` and word frequencies

gamma_draws Data frame (2000 x 2): MCMC draws

theta_est_full Data frame (916 x 2): Full sample topic proportions

theta_est_samp Data frame (916 x 2): Subsample topic proportions

beta_est_full Data frame (2 x 654): Full sample topic-word distributions

beta_est_samp Data frame (2 x 654): Subsample topic-word distributions

lda_data Data frame (916 x 2): LDA validation data

Source

CEO diary data from Bandiera et al (2020), Journal of Political Economy

See Also

`ols_bca_topic`, `ols_bcm_topic`

Examples

```
data(topic_model_data)

# Basic exploration
Y <- topic_model_data$estimation_data$ly
theta <- as.matrix(topic_model_data$theta_est_full)

cat("Sample size:", length(Y), "\n")
cat("Mean log employment:", round(mean(Y), 2), "\n")
cat("Topic 1 mean:", round(mean(theta[, 1]), 3), "\n")
```

Index