Package ‘hbal’

September 10, 2025

Type Package

Title Hierarchically Regularized Entropy Balancing

Version 1.2.15

Date 2025-09-09

Maintainer Yiqing Xu <`yiqingxu@stanford.edu`>

Description Implements hierarchically regularized entropy balancing proposed by Xu and Yang (2022) <doi:10.1017/pan.2022.12>. The method adjusts the covariate distributions of the control group to match those of the treatment group. 'hbal' automatically expands the covariate space to include higher order terms and uses cross-validation to select variable penalties for the balancing conditions.

URL `https://yiqingxu.org/packages/hbal/`

BugReports `https://github.com/xuyiqing/hbal/issues`

License MIT + file LICENSE

Depends R (>= 3.6.0)

Imports Rcpp (>= 1.0.1), estimatr, glmnet, gtable, gridExtra, ggplot2, stringr, nloptr, generics

Suggests MASS, knitr, rmarkdown, broom, ebal

LinkingTo Rcpp, RcppEigen

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation yes

Author Yiqing Xu [aut, cre] (ORCID: <`https://orcid.org/0000-0003-2041-6671`>), Eddie Yang [aut]

Repository CRAN

Date/Publication 2025-09-10 07:50:36 UTC

Contents

`.onAttach` *Subsidiary hbal Function*

Description

Function to load package description.

Usage

```
.onAttach(lib, pkg)
```

Arguments

`lib`	libname
`pkg`	package name

References

Xu, Y., & Yang, E. (2022). Hierarchically Regularized Entropy Balancing. Political Analysis, 1-8. doi:10.1017/pan.2022.12

`att` *Estimating the ATT from an hbal object*

Description

`att` estimates the average treatment effect on the treated (ATT) from an hbal object returned by `hbal`.

Usage

```
att(hbalobject, method="lm_robust", dr=TRUE, displayAll=FALSE, alpha=0.9, ...)
```

Arguments

`hbalobject`	an object of class `hbal` as returned by `hbal`.
`method`	estimation method for the ATT. Default is the Lin (2016) estimator.
`dr`	doubly robust, whether an outcome model is included in estimating the ATT.
`displayAll`	only displays treatment effect by default.
`alpha`	tuning paramter for glmnet
`...`	arguments passed to lm_lin or lm_robust

Details

This is a wrapper for `lm_robust` and `lm_lin` from the **estimatr** package.

Value

A matrix of estimates with their robust standard errors

Author(s)

Yiqing Xu, Eddie Yang

Examples

```
#EXAMPLE 1
set.seed(1984)
N <- 500
X1 <- rnorm(N)
X2 <- rbinom(N,size=1,prob=.5)
X <- cbind(X1, X2)
treat <- rbinom(N, 1, prob=0.5) # Treatment indicator
y <- 0.5 * treat + X[,1] + X[,2] + rnorm(N) # Outcome
dat <- data.frame(treat=treat, X, Y=y)
out <- hbal(Treat = 'treat', X = c('X1', 'X2'), Y = 'Y', data=dat)
sout <- summary(att(out))
```

`contenderJudges` *Data from Black and Owens (2016)*

Description

Data on the contender judges from Black and Owens (2016): Courting the president: how circuit court judges alter their behavior for promotion to the Supreme Court This dataset includes 10,171 period-judge observations for a total of 68 judges. The treatment variable of interest is `treatFinal0`, which indicates whether there was a vacancy in the Supreme Court The outcome of interest is ideological alignment of judges' votes with the sitting President (`presIdeoVote`). The remaining variables are characteristics of the judges and courts, to be used as controls.

Format

A data frame with 10171 rows and 10 columns.

presIdeoVote	ideological alignment of judges' votes with the sitting President (outcome)
treatFinal0	treatment indicator for vacancy period
judgeJCS	judge's Judicial Common Space (JCS)score
presDist	Ideological distribution of the sitting President
panelDistJCS	ideological composition of the panel with whom the judge sat
circmed	median JCS score of the circuit judges
sctmed	JCS score of the median justice on the Supreme Court
coarevtc	indicator for whether the case decision was reversed by the circuit court
casepub	indicator for the publication status of thecourt's opinion
judge	name of the judge

References

- Black, R. C., and Owens, R. J. (2016). Courting the president: how circuit court judges alter their behavior for promotion to the Supreme Court. American Journal of Political Science, 60(1), 30-43.

`covarExclude` *Match Column Names to be Excluded*

Description

Internal function called by `hbal` to serially expand covariates.

Usage

```
covarExclude(colname, exclude)
```

Arguments

`colname`	column name.
`exclude`	list of covariate name pairs or triplets to be excluded.

Value

Logical

Author(s)

Yiqing Xu, Eddie Yang

`covarExpand` *Serial Expansion of Covariates*

Description

Internal function called by `hbal` to serially expand covariates.

Usage

```
covarExpand(X, exp.degree = 3, treatment = NULL, exclude = NULL)
```

Arguments

`X`	matrix of covariates.
`exp.degree`	the degree of the polynomial.
`treatment`	treatment indicator
`exclude`	list of covariate name pairs or triplets to be excluded.

Value

A matrix of serially expanded covariates

Author(s)

Yiqing Xu, Eddie Yang

`crossValidate` *Ridge Penalty Selection through Cross Validation*

Description

Internal function called by `hbal` to select ridge penalties through cross-validation.

Usage

```
crossValidate(
  group.alpha = NULL,
  penalty.pos = NULL,
  penalty.val = NULL,
  group.exact = NULL,
  grouping = NULL,
  folds = NULL,
  treatment = NULL,
  fold.co = NULL,
  fold.tr = NULL,
  coefs = NULL,
  control = NULL,
  constraint.tolerance = NULL,
  print.level = NULL,
  base.weight = NULL,
  full.t = NULL,
  full.c = NULL,
  shuffle.treat = NULL
)
```

Arguments

`group.alpha`	group.alpha. Controls degree of regularization.
`penalty.pos`	positions of user-supplied penalties.
`penalty.val`	values of user-supplied penalties.
`group.exact`	binary indicator of whether each covariate group should be penalized.
`grouping`	different groupings of the covariates.
`folds`	number of folds to perform cross validation.
`treatment`	covariate matrix for treatment group.
`fold.co`	fold assignments for control units.
`fold.tr`	fold assignments for treated units.
`coefs`	starting coefficients (lambda).
`control`	covariate matrix for control group.
`constraint.tolerance`	tolerance level for imbalance.

`print.level`	details of printed output.
`base.weight`	target weight distribution for the control units.
`full.t`	(unresidualized) ovariate matrix for treatment group.
`full.c`	(unresidualized) ovariate matrix for control group.
`shuffle.treat`	whether to create folds for the treated units

Value

group.alpha, lambda

Author(s)

Yiqing Xu, Eddie Yang

`doubleSelection` *Double Selection*

Description

Internal function called by `hbal` to perform double selection.

Usage

```
doubleSelection(X, W, Y, grouping)
```

Arguments

`X`	covaraite matrix
`W`	treatment indicator
`Y`	outcome variable
`grouping`	groupings of covariates

Value

resX, penalty.list, covar.keep

Author(s)

Yiqing Xu, Eddie Yang

`hbal` *Hierarchically Regularized Entropy Balancing*

Description

`hbal` performs hierarchically regularized entropy balancing such that the covariate distributions of the control group match those of the treatment group. `hbal` automatically expands the covariate space to include higher order terms and uses cross-validation to select variable penalties for the balancing conditions.

hbal performs hierarchically regularized entropy balancing such that the covariate distributions of the control group match those of the treatment group. hbal automatically expands the covariate space to include higher order terms and uses cross-validation to select variable penalties for the balancing conditions.

Usage

```
hbal(data, Treat, X, Y = NULL, w = NULL,
     X.expand = NULL, X.keep = NULL, expand.degree = 1,
     coefs = NULL, max.iterations = 200, cv = NULL, folds = 4,
     ds = FALSE, group.exact = NULL, group.alpha = NULL,
     term.alpha = NULL, constraint.tolerance = 1e-3, print.level = 0,
    grouping = NULL, group.labs = NULL, linear.exact = TRUE, shuffle.treat = TRUE,
     exclude = NULL,force = FALSE, seed = 94035)
```

Arguments

`data`	a dataframe that contains the treatment, outcome, and covariates.
`Treat`	a character string of the treatment variable.
`X`	a character vector of covariate names to balance on.
`Y`	a character string of the outcome variable.
`w`	a character string of the weighting variable for base weights
`X.expand`	a character vector of covariate names for serial expansion.
`X.keep`	a character vector of covariate names to keep regardless of whether they are selected in double selection.
`expand.degree`	degree of series expansion. 1 means no expansion. Default is 1.
`coefs`	initial coefficients for the reweighting algorithm (lambdas).
`max.iterations`	maximum number of iterations. Default is 200.
`cv`	whether to use cross validation. Default is `TRUE`.
`folds`	number of folds for cross validation. Only used when cv is `TRUE`.
`ds`	whether to perform double selection prior to balancing. Default is `FALSE`.
`group.exact`	binary indicator of whether each covariate group should be exact balanced.
`group.alpha`	penalty for each covariate group

`term.alpha`	a named vector of user-specified ridge penalties. The names need to be variable names. Value should be non-negative (0 means exact balancing). Only work with ‘expand.degree = 1‘
`constraint.tolerance`	tolerance level for overall imbalance. Default is 1e-3.
`print.level`	details of printed output.
`grouping`	different groupings of the covariates. Must be specified if expand is `FALSE`.
`group.labs`	labels for user-supplied groups
`linear.exact`	seek exact balance on the level terms
`shuffle.treat`	whether to use cross-validation on the treated units. Default is `TRUE`.
`exclude`	list of covariate name pairs or triplets to be excluded.
`force`	binary indicator of whether to expand covariates when there are too many
`seed`	random seed to be set. Set random seed when cv=`TRUE` for reproducibility.

Details

In the simplest set-up, user can just pass in {Treatment, X, Y}. The default settings will serially expand X to include higher order terms, hierarchically residualize these terms, perform double selection to only keep the relevant variables and use cross-validation to select penalities for different groupings of the covariates.

Value

An list object of class `hbal` with the following elements:

`coefs`	vector that contains coefficients from the reweighting algorithm.
`mat`	matrix of serially expanded covariates if expand=`TRUE`. Otherwise, the original covariate matrix is returned.
`penalty`	vector of ridge penalties used for each covariate
`weights`	vector that contains the control group weights assigned by hbal.
`W`	vector of treatment status
`Y`	vector of outcome

Author(s)

Yiqing Xu, Eddie Yang

Yiqing Xu <yiqingxu@stanford.edu>, Eddie Yang <z5yang@ucsd.edu>

References

Xu, Y., & Yang, E. (2022). Hierarchically Regularized Entropy Balancing. Political Analysis, 1-8. doi:10.1017/pan.2022.12

Examples

```
# Example 1
set.seed(1984)
N <- 500
X1 <- rnorm(N)
X2 <- rbinom(N,size=1,prob=.5)
X <- cbind(X1, X2)
treat <- rbinom(N, 1, prob=0.5) # Treatment indicator
y <- 0.5 * treat + X[,1] + X[,2] + rnorm(N) # Outcome
dat <- data.frame(treat=treat, X, Y=y)
out <- hbal(Treat = 'treat', X = c('X1', 'X2'), Y = 'Y', data=dat)
summary(hbal::att(out))

# Example 2
## Simulation from Kang and Shafer (2007).
library(MASS)
set.seed(1984)
n <- 500
X <- mvrnorm(n, mu = rep(0, 4), Sigma = diag(4))
prop <- 1 / (1 + exp(X[,1] - 0.5 * X[,2] + 0.25*X[,3] + 0.1 * X[,4]))
# Treatment indicator
treat <- rbinom(n, 1, prop)
# Outcome
y <- 210 + 27.4*X[,1] + 13.7*X[,2] + 13.7*X[,3] + 13.7*X[,4] + rnorm(n)
# Observed covariates
X.mis <- cbind(exp(X[,1]/2), X[,2]*(1+exp(X[,1]))^(-1)+10,
    (X[,1]*X[,3]/25+.6)^3, (X[,2]+X[,4]+20)^2)
dat <- data.frame(treat=treat, X.mis, Y=y)
out <- hbal(Treat = 'treat', X = c('X1', 'X2', 'X3', 'X4'), Y='Y', data=dat)
summary(att(out))
```

`hbal-data` *Data from Black and Owens (2016) and Hazlett (2020)*

Description

The contenderJudges dataset is from Black and Owens (2016): Courting the president: how circuit court judges alter their behavior for promotion to the Supreme Court This dataset includes 10,171 period-judge observations for a total of 68 judges. The treatment variable of interest is `treatFinal0`, which indicates whether there was a vacancy in the Supreme Court The outcome of interest is ideological alignment of judges' votes with the sitting President (`presIdeoVote`). The remaining variables are characteristics of the judges and courts, to be used as controls.

The LaLonde dataset has treated units from Dehejia and Wahba (1999), containing 185 individuals; data on the control units is from Panel Study of Income Dynamics (PSID-1), containing 2,490 individuals.

Usage

```
data(hbal)
```

Source

- Black, R. C., and Owens, R. J. (2016). Courting the president: how circuit court judges alter their behavior for promotion to the Supreme Court. American Journal of Political Science, 60(1), 30-43.
- Dehejia, R. H., and Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. Journal of the American statistical Association, 94(448), 1053-1062.
- Hazlett, C. (2020). KERNEL BALANCING. Statistica Sinica, 30(3), 1155-1189.

`lalonde` *Data from Hazlett (2020)*

Description

Data on the treated units is from Dehejia and Wahba (1999), containing 185 individuals; data on the control units is from Panel Study of Income Dynamics (PSID-1), containing 2,490 individuals.

Format

A data frame with 2675 rows and 13 columns.

nsw treatment indicator of whether an individual participated in the National Supported Work (NSW) program

age

educ years of education

black demographic indicator variables for Black

hisp idemographic indicator variables for Hispanic

married demographic indicator variables for married

re74 real earnings in 1974

re75 real earnings in 1975

re78 real earnings in 1978, outcome

u74 unemployment indicator for 1974

u75 unemployment indicator for 1975

u78 unemployment indicator for 1978

nodegr indicator for no high school degree

References

- Dehejia, R. H., and Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. Journal of the American statistical Association, 94(448), 1053-1062.
- Hazlett, C. (2020). KERNEL BALANCING. Statistica Sinica, 30(3), 1155-1189.

`plot.hbal` *Plotting Covariate Balance from an* `hbal` *Object*

Description

This function plots the covariate difference between the control and treatment groups in standardized means before and after weighting.

Usage

```
## S3 method for class 'hbal'
plot(x, type = 'balance', log = TRUE, base_size = 10, ...)
```

Arguments

`x`	an object of class `hbalobject` as returned by `hbal`.
`type`	type of graph to plot.
`log`	log scale for the weight plot
`base_size`	base font size
`...`	Further arguments to be passed to `plot.hbal()`.

Value

A matrix of ggplots of covariate balance by group

Author(s)

Yiqing Xu, Eddie Yang

`summary.hbal` *Summarizing from an* `hbal` *Object*

Description

This function prints a summary from an `hbal` Object.

Usage

```
## S3 method for class 'hbal'
summary(object, print.level = 0, ...)
```

Arguments

`object`	an object of class `hbalobject` as returned by `hbal`.
`print.level`	level of details to be printed
`...`	Further arguments to be passed to `summary.hbal()`.

Value

a summary table

Author(s)

Yiqing Xu, Eddie Yang

`updateCoef` *Update lambda*

Description

Internal function called by `hbal` to residualize covariates.

Usage

```
updateCoef(old.coef, new.coef, counter)
```

Arguments

`old.coef`	previous coefficients
`new.coef`	new coefficients
`counter`	which fold in CV

Value

updated coefficients

Author(s)

Yiqing Xu, Eddie Yang

Index