

# Package ‘hbal’

September 17, 2026

**Type** Package

**Title** Hierarchically Regularized Entropy Balancing

**Version** 1.3.0

**Date** 2026-09-17

**Description** Implements hierarchically regularized entropy balancing proposed by Xu and Yang (2022) <[doi:10.1017/pan.2022.12](https://doi.org/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, testthat (>= 3.0.0)

**LinkingTo** Rcpp, RcppEigen

**Encoding** UTF-8

**LazyData** true

**RoxxygenNote** 7.2.3

**Config/testthat/edition** 3

**NeedsCompilation** yes

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

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

**Repository** CRAN

**Date/Publication** 2026-09-17 15:20:02 UTC

Contents

|                           |           |
|---------------------------|-----------|
| .onAttach . . . . .       | 2         |
| att . . . . .             | 3         |
| balanceData . . . . .     | 6         |
| contenderJudges . . . . . | 7         |
| covarExclude . . . . .    | 8         |
| covarExpand . . . . .     | 8         |
| crossValidate . . . . .   | 9         |
| doubleSelection . . . . . | 10        |
| hbal . . . . .            | 11        |
| hbal-data . . . . .       | 14        |
| lalonde . . . . .         | 15        |
| plot.hbal . . . . .       | 16        |
| summary.hbal . . . . .    | 16        |
| updateCoef . . . . .      | 17        |
| <b>Index</b>              | <b>18</b> |

---

|           |                                 |
|-----------|---------------------------------|
| .onAttach | <i>Subsidiary hbal Function</i> |
|-----------|---------------------------------|

---

Description

Function to load package description.

Usage

.onAttach(lib, pkg)

Arguments

|     |              |
|-----|--------------|
| lib | libname      |
| pkg | package name |

Value

No return value, called for the side effect of printing the package startup message.

References

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

---

|     |                                               |
|-----|-----------------------------------------------|
| att | <i>Estimating the ATT from an hbal object</i> |
|-----|-----------------------------------------------|

---

## Description

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

## Usage

```
att(hbalobject, method="abw", dr=TRUE, displayAll=FALSE, alpha=0.9,
    seed=NULL, nfolds=5, ...)
```

## Arguments

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| hbalobject | an object of class hbal as returned by hbal.                                                                                                                                                                                                                                                                                                                                                                                               |
| method     | estimation method for the ATT. The default "abw" (Augmented Balancing Weights; new in version 1.3.0) is a cross-fitted, Neyman-orthogonal estimator that combines the hbal weights with a ridge-regularized control-group outcome model (see Details). "lm_robust" reproduces the default of versions before 1.3.0 (weighted regression via lm_robust); "lm_lin" (Lin 2013) and "elnet" (Athey, Imbens and Wager 2018) are also available. |
| dr         | doubly robust, whether an outcome model is included in estimating the ATT. With dr = FALSE every method returns the weighted difference in means. "abw" and "lm_robust" compute it directly; "lm_lin" and "elnet" are defined only with an outcome model, so att estimates them as "lm_robust", giving the same estimate, standard error and degrees of freedom as att(x, method = "lm_robust", dr = FALSE).                               |
| displayAll | only displays treatment effect by default. If TRUE, returns the fitted lm_robust or lm_lin object, or, for method = "abw", a list with the estimate and its components (see Value).                                                                                                                                                                                                                                                        |
| alpha      | tuning parameter for glmnet (method = "elnet" only).                                                                                                                                                                                                                                                                                                                                                                                       |
| seed       | a single number that fixes the cross-fitting fold assignment of method = "abw"; NULL (the default) assigns folds in the row order of the controls. The assignment is a deterministic function of seed; set.seed() is never called. Ignored by the other methods.                                                                                                                                                                           |
| nfolds     | number of cross-fitting folds for method = "abw". Default is 5. Reduced automatically, with a message, when the control group is small relative to the number of covariates (see Details). Ignored by the other methods.                                                                                                                                                                                                                   |
| ...        | arguments passed to lm_lin or lm_robust (e.g. se_type, clusters). Not accepted by method = "abw".                                                                                                                                                                                                                                                                                                                                          |

## Details

The default method = "abw" (Augmented Balancing Weights; Ben-Michael, Feller, Hirshberg and Zubizarreta 2021; Bruns-Smith, Dukes, Feller and Ogburn 2023) combines the hbal weights with a control-group outcome model in an augmented moment condition that is Neyman-orthogonal: its first-order sensitivity to estimation error in either nuisance component (the outcome model, given hbal's balance on mat; the weights, given a correct outcome model) is zero. Let  $w_i$  be the base weights of the treated units,  $W_1$  their sum,  $\gamma_i$  the hbal weights of the controls (weights.co, which also sum to  $W_1$ ), and  $\hat{\mu}_0$  a ridge regression of the outcome on the columns of hbalobject\$mat, fitted on the controls only and weighted by their  $\gamma_i$ : every column is standardized with the mean and standard deviation of the controls used in the fit, the intercept is not penalized, and the penalty  $\lambda$  is chosen once per call from the full control sample by generalized cross-validation (GCV; Golub, Heath and Wahba 1979) over a fixed grid of candidate values, restricted to candidates whose effective degrees of freedom do not exceed  $\min(p, n_0/2) + 1$ , with  $n_0$  controls and  $p$  columns in mat, so that the fit can never approach an interpolation of the controls. The selection is deterministic: no data splitting and no random numbers are involved, and  $\lambda = 0$  (weighted least squares) is chosen whenever it is admissible and minimizes the GCV score. The estimate is

$$\hat{\tau} = \frac{1}{W_1} \left[ \sum_{T_i=1} w_i (Y_i - \hat{m}_i) - \sum_{T_i=0} \gamma_i \hat{e}_i \right],$$

where  $\hat{m}_i$  is the outcome model fitted on all controls and evaluated at treated unit  $i$ , and  $\hat{e}_i$  is the cross-fitted residual of control  $i$ : the controls are split into nfolds folds and each control's residual uses the fit obtained without its own fold. The number of folds actually used is  $K = \max(1, \min(\text{nfolds}, \lfloor n_0/(2p) \rfloor))$ , ( $n_0$  and  $p$  as above);  $K = 1$  means no cross-fitting. Setting nfolds = 1 disables cross-fitting (the correction term above becomes identically zero) and lowers root-mean-squared error at small sample sizes in simulations accompanying this release, at the cost of an understated standard error and, under outcome-model misspecification, added bias; nfolds = 5 (the default) is recommended unless a smaller root-mean-squared error is wanted more than a reliable standard error. The standard error is influence-function based: with  $\psi_i = w_i T_i (Y_i - \hat{m}_i - \hat{\tau}) - \gamma_i (1 - T_i) \hat{e}_i$ , the variance estimate is  $\sum_i \psi_i^2 / W_1^2$ , treating the weights and the outcome fit as fixed; p-values and the 95 percent confidence interval use a t distribution with  $n - 1$  degrees of freedom. With dr = FALSE the outcome model is dropped ( $\hat{m}_i = 0$ ,  $\hat{e}_i = Y_i$ ) and the same formulas give the weighted difference in means. Fold assignment is deterministic: repeated calls with the same seed (including NULL) return identical results. If the entropy-balancing weights of hbalobject did not converge (hbalobject\$converged is FALSE), att issues a warning for either value of dr, because the estimate and its standard error may then be unreliable.

method = "lm\_robust" (the default before version 1.3.0) and method = "lm\_lin" are wrappers for lm\_robust and lm\_lin from the **estimatr** package, fitted with the hbal weights; method = "elnet" implements the approximate residual balancing estimator of Athey, Imbens and Wager (2018) via **glmnet**. For method = "lm\_robust", the default se\_type understates the standard error by 5 to 7 percent in the same simulations; se\_type = "HC3" corrects this for samples of 500 or fewer but still runs about 5 percent short at 1,000. Before version 1.3.0, dr = FALSE was accepted and silently ignored by method = "lm\_lin" and method = "elnet"; both now fall back to "lm\_robust".

## Value

A data frame with one row and seven columns (Estimate, Std. Error, t value, Pr(>|t|), CI Lower, CI Upper, DF) when displayAll = FALSE (the default) or method = "elnet". When displayAll

= TRUE: for method = "lm\_robust" or "lm\_lin", the fitted model object returned by `lm_robust` or `lm_lin`; for method = "abw", a plain list with elements

**estimate** the ATT point estimate.

**se** its influence-function standard error.

**df** degrees of freedom of the t distribution used for the p-value and the confidence interval,  $n - 1$ .

**method** "abw".

**dr** the dr value used.

**influence** numeric vector of length  $n$ : the per-unit moment contributions  $\psi_i$  (see Details), in the row order of `hbalobject`; they sum to zero.

**fold** integer vector of length  $n$ : the cross-fitting fold of each control (NA for treated units, and for every unit when `dr = FALSE`).

**nfolds** the number of folds actually used (NA when `dr = FALSE`).

**nuisance** a list with `coef_full` (coefficients of the ridge outcome model fitted on all controls, on the original scale of `mat`, named "(Intercept)" followed by the columns of `mat`), `coef_folds` (a list of the `nfolds` fold-specific coefficient vectors), and `rank_full` (the rank of the full-control design including the intercept); NULL, `list()` and NA respectively when `dr = FALSE`.

**weights** a list with `treated` (the base weights of the treated units) and `control` (the `hbal` weights of the controls), each in the row order of `hbalobject$mat` within its group.

## Author(s)

Yiqing Xu, Eddie Yang

## References

- Ben-Michael, E., Feller, A., Hirshberg, D. A., and Zubizarreta, J. R. (2021). The balancing act in causal inference. *arXiv:2110.14831*.
- Bruns-Smith, D., Dukes, O., Feller, A., and Ogburn, E. L. (2023). Augmented balancing weights as linear regression. *arXiv:2304.14545*.
- Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2), 215-223.
- Athey, S., Imbens, G. W., and Wager, S. (2018). Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society: Series B*, 80(4), 597-623.
- Lin, W. (2013). Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique. *The Annals of Applied Statistics*, 7(1), 295-318.

## 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))      # default: augmented balancing weights (abw)
att(out, method = "lm_robust") # the default before version 1.3.0
```

balanceData

*Balance Statistics from an hbal Object as a Data Frame***Description**

balanceData returns the covariate balance statistics of an hbal object in long form, with a column giving each term's covariate group. This is the data plot draws for type = "balance", in a shape a **ggplot2** call can consume directly.

**Usage**

```
balanceData(hbalobject)
```

**Arguments**

hbalobject      an object of class hbal as returned by hbal.

**Details**

The returned frame has two rows per column of hbalobject\$mat: one before weighting and one after. The values come from hbalobject\$bal.tab and are therefore rounded to two decimals, exactly as summary and plot report them. covar.group repeats each group label of hbalobject\$grouping as many times as that group has columns of mat, in the order the groups appear.

**Value**

A data frame with  $2p$  rows, where  $p$  is the number of columns of hbalobject\$mat, and the columns

**term** character; the column name of mat the row refers to.

**covar.group** factor; the covariate group of that term. The levels are the names of hbalobject\$grouping, in that order.

**adjustment** factor with levels "before" and "after", in that order.

**std.diff** numeric; the standardized difference in means between the treated and the control group, before or after weighting.

**tr.mean, co.mean, w.co.mean** numeric; the treated mean, the unweighted control mean and the weighted control mean of the term. Each is repeated on both of the term's rows.

**Author(s)**

Yiqing Xu, Eddie Yang

## Examples

```
#EXAMPLE
set.seed(1984)
N <- 500
X1 <- rnorm(N)
X2 <- rbinom(N, size = 1, prob = .5)
treat <- rbinom(N, 1, prob = 0.5)
Y <- 0.5 * treat + X1 + X2 + rnorm(N)
dat <- data.frame(treat = treat, X1 = X1, X2 = X2, Y = Y)
out <- hbal(Treat = 'treat', X = c('X1', 'X2'), Y = 'Y', data = dat)
balanceData(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 the court's opinion

**judge** name of the judge

## Source

Black, R. C., and Owens, R. J. Replication data for: Courting the President: How Circuit Court Judges Alter Their Behavior for Promotion to the Supreme Court. Harvard Dataverse, [doi:10.7910/DVN/25302](https://doi.org/10.7910/DVN/25302). That deposit is released under the Creative Commons CC0 1.0 Universal Public Domain Dedication.

## 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 | <i>Match Column Names to be Excluded</i> |
|--------------|------------------------------------------|

---

## 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 | <i>Serial Expansion of Covariates</i> |
|-------------|---------------------------------------|

---

## 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 list with two elements: `mat`, the matrix of serially expanded covariates, and `grouping`, an integer vector giving the number of columns of `mat` in each term group (three groups when `exp.degree = 2`, six when `exp.degree = 3`).

**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**

|                          |                                                                       |
|--------------------------|-----------------------------------------------------------------------|
| <code>group.alpha</code> | <code>group.alpha</code> . Controls degree of regularization.         |
| <code>penalty.pos</code> | positions of user-supplied penalties.                                 |
| <code>penalty.val</code> | values of user-supplied penalties.                                    |
| <code>group.exact</code> | binary indicator of whether each covariate group should be penalized. |
| <code>grouping</code>    | different groupings of the covariates.                                |

|                                   |                                                      |
|-----------------------------------|------------------------------------------------------|
| <code>folds</code>                | number of folds to perform cross validation.         |
| <code>treatment</code>            | covariate matrix for treatment group.                |
| <code>fold.co</code>              | fold assignments for control units.                  |
| <code>fold.tr</code>              | fold assignments for treated units.                  |
| <code>coefs</code>                | starting coefficients (lambda).                      |
| <code>control</code>              | covariate matrix for control group.                  |
| <code>constraint.tolerance</code> | tolerance level for imbalance.                       |
| <code>print.level</code>          | details of printed output.                           |
| <code>base.weight</code>          | target weight distribution for the control units.    |
| <code>full.t</code>               | (unresidualized) oariate matrix for treatment group. |
| <code>full.c</code>               | (unresidualized) oariate matrix for control group.   |
| <code>shuffle.treat</code>        | whether to create folds for the treated units        |

**Value**

A single numeric value: the mean cross-validation loss at the supplied penalties, or Inf when that mean is not finite. `hbal` minimizes it over the penalties with `nloptr`.

**Author(s)**

Yiqing Xu, Eddie Yang

---

|                              |                         |
|------------------------------|-------------------------|
| <code>doubleSelection</code> | <i>Double Selection</i> |
|------------------------------|-------------------------|

---

**Description**

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

**Usage**

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

**Arguments**

|                       |                         |
|-----------------------|-------------------------|
| <code>X</code>        | covariate matrix        |
| <code>W</code>        | treatment indicator     |
| <code>Y</code>        | outcome variable        |
| <code>grouping</code> | 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 = NULL)
```

**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 to select the ridge penalties. The default, NULL, is treated as FALSE: no cross-validation is run and, unless group.alpha is supplied, every penalty is 0, which asks for exact balance. Set cv = TRUE to choose the penalties by cross-validation. |

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>folds</code>                | number of folds for cross validation. Only used when <code>cv</code> is <code>TRUE</code> .                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>ds</code>                   | whether to perform double selection prior to balancing. Default is <code>FALSE</code> .                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>group.exact</code>          | binary indicator of whether each covariate group should be exact balanced.                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>group.alpha</code>          | penalty for each covariate group                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>term.alpha</code>           | 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 <code>'expand.degree = 1'</code>                                                                                                                                                                                                                                                                                        |
| <code>constraint.tolerance</code> | tolerance level for overall imbalance. Default is <code>1e-3</code> .                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>print.level</code>          | details of printed output.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>grouping</code>             | different groupings of the covariates. Must be specified if <code>expand</code> is <code>FALSE</code> .                                                                                                                                                                                                                                                                                                                                                                               |
| <code>group.labs</code>           | labels for user-supplied groups                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>linear.exact</code>         | seek exact balance on the level terms                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>shuffle.treat</code>        | whether to use cross-validation on the treated units. Default is <code>TRUE</code> .                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>exclude</code>              | list of covariate name pairs or triplets to be excluded.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>force</code>                | binary indicator of whether to expand covariates when there are too many                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>seed</code>                 | random seed passed to <code>set.seed()</code> at the start of the cross-validation when <code>cv = TRUE</code> . The default, <code>NULL</code> , leaves the random number generator untouched, so repeated cross-validated fits can differ. Supply a seed for reproducible results. <code>seed = 94035</code> was the default in <code>hbal 1.2.15</code> and earlier and reproduces the cross-validated results of those versions exactly. Ignored when no cross-validation is run. |

## Details

In the simplest set-up, the user can just pass in `{Treat, X, Y}`. With the default settings `hbal` seeks exact balance on the covariates as supplied: there is no series expansion (`expand.degree = 1`), no double selection (`ds = FALSE`) and no cross-validation (`cv` resolves to `FALSE`), and every ridge penalty is 0. Set `expand.degree` to 2 or 3 to serially expand `X` to include higher order terms and hierarchically residualize them, `ds = TRUE` to perform double selection and keep only the relevant variables, and `cv = TRUE` to select penalties for the different groupings of the covariates by cross-validation.

## Value

A list object of class `hbal` with the elements below. Rows of data with a missing value in the treatment, the outcome, any covariate or the weighting variable are dropped before estimation, so every per-unit element has one entry per retained row, in the order of data.

|                         |                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>converged</code>  | integer, 1 if the entropy-balancing algorithm converged within <code>max.iterations</code> and 0 otherwise. <code>att</code> warns when it is 0. |
| <code>weights</code>    | numeric vector over all units: the <code>hbal</code> weight of each control unit and the base weight of each treated unit.                       |
| <code>weights.co</code> | numeric vector over the control units: their entropy-balancing weights, normalized to sum to the total base weight of the treated units.         |

|               |                                                                                                                                                                                                                                                                                |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coefs         | numeric vector of the Lagrangian multipliers returned by the reweighting algorithm, one per column of <code>mat</code> plus one for the normalizing constraint.                                                                                                                |
| Treatment     | numeric vector of the treatment indicator, 1 for treated and 0 for control.                                                                                                                                                                                                    |
| mat           | numeric matrix of the covariates actually balanced on, on their original scale, after series expansion ( <code>expand.degree</code> ), removal of collinear columns and double selection ( <code>ds</code> ). Its column names are the covariate names with a position suffix. |
| grouping      | named numeric vector giving the number of columns of <code>mat</code> in each covariate group; the names are the group labels.                                                                                                                                                 |
| group.penalty | named numeric vector with one ridge penalty per covariate group: chosen by cross-validation when <code>cv = TRUE</code> , taken from <code>group.alpha</code> when that is supplied, and 0 otherwise.                                                                          |
| term.penalty  | named numeric vector with one ridge penalty per column of <code>mat</code> .                                                                                                                                                                                                   |
| bal.tab       | numeric matrix with one row per column of <code>mat</code> and the columns <code>Tr.Mean</code> , <code>Co.Mean</code> , <code>W.Co.Mean</code> , <code>Std.Diff.(O)</code> and <code>Std.Diff.(W)</code> , rounded to two decimals.                                           |
| base.weights  | numeric vector of the base weights: the variable named by <code>w</code> , or 1 for every unit when <code>w</code> is <code>NULL</code> .                                                                                                                                      |
| Treat         | character string, the name of the treatment variable.                                                                                                                                                                                                                          |
| Outcome       | numeric vector of the outcome. Present only when <code>Y</code> is supplied.                                                                                                                                                                                                   |
| Y             | character string, the name of the outcome variable. Present only when <code>Y</code> is supplied.                                                                                                                                                                              |
| call          | the matched call.                                                                                                                                                                                                                                                              |

### 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).
if (requireNamespace("MASS", quietly = TRUE)) {
  set.seed(1984)
  n <- 500
  X <- MASS::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.

lalonge

*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

**Source**

National Supported Work (NSW) demonstration data from LaLonde (1986), with the treated units as re-analyzed by Dehejia and Wahba (1999) and PSID-1 controls. Taken from the replication archive of Xu and Yang (2022): Xu, Y., and Yang, E. Replication Data for: Hierarchically Regularized Entropy Balancing. Harvard Dataverse, [doi:10.7910/DVN/QI2WP9](https://doi.org/10.7910/DVN/QI2WP9). That archive is released under the Creative Commons CC0 1.0 Universal Public Domain Dedication.

**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 | <i>Plotting Covariate Balance from an hbal Object</i> |
|-----------|-------------------------------------------------------|

---

**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**

For type = "weight", a ggplot object holding the histogram and density of the control units' balancing weights (on the log scale when log = TRUE); it is drawn when printed. For any other type, the covariate-balance panels are drawn on the current graphics device and the assembled gtable is returned invisibly.

**Author(s)**

Yiqing Xu, Eddie Yang

---

|              |                                        |
|--------------|----------------------------------------|
| summary.hbal | <i>Summarizing from an hbal Object</i> |
|--------------|----------------------------------------|

---

**Description**

This function prints a summary from an hbal Object.

**Usage**

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

**Arguments**

|                          |                                                                               |
|--------------------------|-------------------------------------------------------------------------------|
| <code>object</code>      | an object of class <code>hbalobject</code> as returned by <code>hbal</code> . |
| <code>print.level</code> | level of details to be printed                                                |
| <code>...</code>         | Further arguments to be passed to <code>summary.hbal()</code> .               |

**Value**

Called for its side effect of printing a summary of the `hbal` object: the call, the numbers of treated and control units, the covariate groups with their penalties, and the balance table. The balance table (`object$bal.tab`) is returned invisibly.

**Author(s)**

Yiqing Xu, Eddie Yang

---

|                         |                      |
|-------------------------|----------------------|
| <code>updateCoef</code> | <i>Update lambda</i> |
|-------------------------|----------------------|

---

**Description**

Internal function called by `hbal` to residualize covariates.

**Usage**

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

**Arguments**

|                       |                       |
|-----------------------|-----------------------|
| <code>old.coef</code> | previous coefficients |
| <code>new.coef</code> | new coefficients      |
| <code>counter</code>  | which fold in CV      |

**Value**

updated coefficients

**Author(s)**

Yiqing Xu, Eddie Yang

# Index

## \* datasets

contenderJudges, [7](#)

hbal-data, [14](#)

lalonge, [15](#)

.onAttach, [2](#)

att, [3](#)

balanceData, [6](#)

contenderJudges, [7](#)

covarExclude, [8](#)

covarExpand, [8](#)

CrossValidate (crossValidate), [9](#)

crossValidate, [9](#)

doubleSelection, [10](#)

hbal, [11](#)

hbal-data, [14](#)

lalonge, [15](#)

plot (plot.hbal), [16](#)

plot.hbal, [16](#)

summary (summary.hbal), [16](#)

summary.hbal, [16](#)

updateCoef, [17](#)