Package 'MatchLinReg'

May 7, 2026

Type Package

Title Combining Matching and Linear Regression for Causal Inference

Version 0.8.1

Date 2022-08-30

Author Alireza S. Mahani, Mansour T.A. Sharabiani

Maintainer Alireza S. Mahani <alireza.s.mahani@gmail.com>

Description Core functions as well as diagnostic and calibration tools for combining matching and linear regression for causal inference in observational studies.

License GPL (>= 2)

Depends R (>= 3.1.0)

Imports Hmisc, Matching, graphics, stats

NeedsCompilation no

Repository CRAN

Suggests R.rsp

VignetteBuilder R.rsp

Date/Publication 2022-08-30 13:30:08 UTC

Contents

`lalonde` *Lalonde's National Supported Work Demonstration data*

Description

One of the datasets used by Dehejia and Wahba in their paper "Causal Effects in Non-Experimental Studies: Reevaluating the Evaluation of Training Programs." Also used as an example dataset in the MatchIt package.

Usage

```
data("lalonde")
```

Format

A data frame with 614 observations on the following 10 variables.

`treat` treatment indicator; 1 if treated in the National Supported Work Demonstration, 0 if from the Current Population Survey

`age` age, a numeric vector.

`educ` years of education, a numeric vector between 0 and 18.

`black` a binary vector, 1 if `black`, 0 otherwise.

`hispan` a binary vector, 1 if `hispanic`, 0 otherwise.

`married` a binary vector, 1 if `married`, 0 otherwise.

`nodegree` a binary vector, 1 if no degree, 0 otherwise.

`re74` earnings in 1974, a numeric vector.

`re75` earnings in 1975, a numeric vector.

`re78` earnings in 1978, a numeric vector (outcome variable).

Details

This data set has been taken from `twang` package, with small changes to field descriptions.

Source

http://www.columbia.edu/~rd247/nswdata.html http://cran.r-project.org/src/contrib/Descriptions/MatchIt.html

References

Lalonde, R. (1986). Evaluating the econometric evaluations of training programs with experimental data. American Economic Review 76: 604-620.

Dehejia, R.H. and Wahba, S. (1999). Causal Effects in Nonexperimental Studies: Re-Evaluating the Evaluation of Training Programs. Journal of the American Statistical Association 94: 1053-1062.

lindner *Lindner Center data on 996 PCI patients analyzed by Kereiakes et al. (2000)*

Description

These data are adapted from the lindner dataset in the USPS package. The description comes from that package, except for the variable sixMonthSurvive, which is a recode of lifepres

Data from an observational study of 996 patients receiving an initial Percutaneous Coronary Intervention (PCI) at Ohio Heart Health, Christ Hospital, Cincinnati in 1997 and followed for at least 6 months by the staff of the Lindner Center. The patients thought to be more severely diseased were assigned to treatment with abciximab (an expensive, high-molecular-weight IIb/IIIa cascade blocker); in fact, only 298 (29.9 percent) of patients received usual-care-alone with their initial PCI.

Usage

```
data("lindner")
```

Format

A data frame of 10 variables collected on 996 patients; no NAs.

`lifepres` Mean life years preserved due to survival for at least 6 months following PCI; numeric value of either 11.4 or 0.

`cardbill` Cardiac related costs incurred within 6 months of patient's initial PCI; numeric value in 1998 dollars; costs were truncated by death for the 26 patients with lifepres == 0.

`abcix` Numeric treatment selection indicator; 0 implies usual PCI care alone; 1 implies usual PCI care deliberately augmented by either planned or rescue treatment with abciximab.

`stent` Coronary stent deployment; numeric, with 1 meaning YES and 0 meaning NO.

`height` Height in centimeters; numeric integer from 108 to 196.

`female` Female gender; numeric, with 1 meaning YES and 0 meaning NO.

`diabetic` Diabetes mellitus diagnosis; numeric, with 1 meaning YES and 0 meaning NO.

`acutemi` Acute myocardial infarction within the previous 7 days; numeric, with 1 meaning YES and 0 meaning NO.

`ejecfrac` Left ejection fraction; numeric value from 0 percent to 90 percent.

`ves1proc` Number of vessels involved in the patient's initial PCI procedure; numeric integer from 0 to 5.

`sixMonthSurvive` Survival at six months - a recoded version of lifepres.

Details

This data set and documentation is taken from `twang` package.

References

Kereiakes DJ, Obenchain RL, Barber BL, et al. Abciximab provides cost effective survival advantage in high volume interventional practice. Am Heart J 2000; 140: 603-610.

`mlr` *Creating a series of matched data sets with different calibration parameters*

Description

Creating a series of matched data sets with different calibration parameters. The output of this function can be supplied to `summary.mlr` and then `plot.summary.mlr` methods to generate diagnostic and calibration plots.

Usage

```
mlr(tr, Z.i = NULL, Z.o = mlr.generate.Z.o(Z.i), psm = TRUE
  , caliper.vec = c(0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 5, Inf)
  , ...)
```

Arguments

`tr`	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
`Z.i`	Matrix of adjustment covariates included in linear regression. We must have `nrow(Z.i) == length(tr)`.
`Z.o`	Matrix of adjustment covariates (present in generative model but) omitted from regression estimation. We must have `nrow(Z.o) == length(tr)`.
`psm`	Boolean flag, indicating whether propensity score matching should be used (`TRUE`) or Mahalanobis matching (`FALSE`).
`caliper.vec`	Vector of matching calipers used.
`...`	Other parameters passed to `mlr.match`.

Value

A list with the following fields:

`tr`	Same as input.
`Z.i`	Same as input.
`Z.o`	Same as input.
`idx.list`	List of observation indexes for each matched data set.
`caliper.vec`	Same as input.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

`mlr.bias` *Treatment effect bias*

Description

Calculating treatment effect bias due to misspecified regression, using coefficients of omitted covariates (if supplied) or a constrained bias estimation approach.

Usage

```
mlr.bias(tr, Z.i = NULL, Z.o, gamma.o = NULL
  , idx = 1:length(tr))
```

Arguments

`tr`	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
`Z.i`	Matrix of adjustment covariates included in linear regression. We must have `nrow(Z.i) == length(tr)`.
`Z.o`	Matrix of adjustment covariates (present in generative model but) omitted from regression estimation. We must have `nrow(Z.o) == length(tr)`.
`gamma.o`	Vector of coefficients for omitted adjustment covariates.
`idx`	Index of observations to be used, with possible duplication, e.g. as indexes of matched subset.

Details

For `single`, `subspace` and `absolute`, biases are calculated using the constrained bias estimation framework, i.e. L2 norm of `Z.o%*%gamma.o` is taken to be `length(tr)` (mean squared of 1).

Value

A list with the following elements is returned:

`gamma.o`	If function argument `gamma.o` is `NULL`, this field will be `NA`. Otherwise, this will be the covariate omission bias for the given coefficient values.
`single`	A list with elements: 1) `bias`: bias for the omitted covariate with maximum absolute bias, 2) `bias.vec`: vector of biases for all omitted covariates, 3) `dir`: vector of length `length(tr)`, being the particular column of `Z.o` with maximum absolute bias (after orthogonalization and normalization), 4) `idx`: column number for `Z.o` corresponding to `dir`.

subspace	A list with elements: 1) bias: bias in direction within omitted covariate subspace with maximum absolute bias, 2) dir: direction in omitted-covariate subspace (and orthogonal to subspace spanned by {1,Z.i}) corresponding to the bias in previous element.
absolute	A list with elements: 1) bias: bias in direction within subspace orthogonal to {1,Z.i} with maximum absolute bias, 2) dir: direction in aforementioned subspace corresponding to maximum absolute bias.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

Examples

```
# number of included adjustment covariates
K <- 10
# number of observations in treatment group
Nt <- 100
# number of observations in control group
Nc <- 100
N <- Nt + Nc
# number of omitted covariates
Ko <- 3

# treatment indicator variable
tr <- c(rep(1, Nt), rep(0, Nc))
# matrix of included (adjustment) covariates
Z.i <- matrix(runif(K*N), ncol = K)
# matrix of omitted covariates
Z.o <- matrix(runif(Ko*N), ncol = Ko)
# coefficients of omitted covariates
gamma.o <- runif(Ko)

retobj <- mlr.bias(tr = tr, Z.i = Z.i, Z.o = Z.o, gamma.o = gamma.o)

# 1) using actual coefficients for computing bias
ret <- retobj$gamma.o

# comparing with brute-force approach
X.i <- cbind(tr, 1, Z.i)
ret2 <- (solve(t(X.i) %*% X.i, t(X.i) %*% Z.o %*% gamma.o))[1]

cat("check 1:", all.equal(ret2, ret), "\n")

# comparing with single method
Z.o.proj <- mlr.orthogonalize(X = cbind(1, Z.i), Z = Z.o, normalize = TRUE)
ret3 <- (solve(t(X.i) %*% X.i, t(X.i) %*% Z.o.proj))[1, ]
```

```
cat("check 2:", all.equal(ret3, retobj$single$bias.vec), "\n")

ret4 <- (solve(t(X.i) %*% X.i, t(X.i) %*% retobj$subspace$dir))[1, ]

cat("check 3:", all.equal(as.numeric(ret4), as.numeric(retobj$subspace$bias)), "\n")

ret4 <- (solve(t(X.i) %*% X.i, t(X.i) %*% retobj$absolute$dir))[1, ]

cat("check 4:", all.equal(as.numeric(ret4), as.numeric(retobj$absolute$bias)), "\n")
```

`mlr.bias.constructor` *Generating the treatment effect bias constructor vector*

Description

Generaring the vector that, multiplied by `Z.o%*%gamma.o` (contribution of omitted covariates to outcome), produces the treatment effect bias - due to model misspecification in the form of covariate omission - when using linear regression for causal inference.

Usage

```
mlr.bias.constructor(tr, Z.i = NULL, details = FALSE, idx = 1:length(tr))
```

Arguments

`tr`	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
`Z.i`	Matrix of adjustment covariates included in linear regression. We must have `nrow(Z.i) == length(tr)`.
`details`	Boolean flag, indicating whether intermediate objects used in generating the constrcutor vector must be returned or not. This only works if at least one adjustment covariate is included in the regression (`Z.i` is not `NULL`), and there are no repeated observations, i.e. `max(table(idx))==1`.
`idx`	Index of observations to be used, with possible duplication, e.g. as indexes of matched subset.

Value

A vector of same length as `tr` is returned. If `details = TRUE` and `Z.i` is not `NULL`, then the following objects are attached as attributes:

`p`	Vector of length `ncol(Z.i)`, reflecting the sum of each included covariate in treatment group.
`q`	Vector of length `ncol(Z.i)`, reflecting the sum of each included covariate across both treatment and control groups.

`u.i`	Vector of length `ncol(Z.i)`, reflecting the mean difference between groups (control - treatment) for each included covariate.
`A`	Weighted, within-group covariance matrix of included covariates. It is a square matrix of dimension `ncol(Z.i)`.
`iA`	Inverse of `A`.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

Examples

```
# number of included adjustment covariates
K <- 10
# number of observations in treatment group
Nt <- 100
# number of observations in control group
Nc <- 100
N <- Nt + Nc

# treatment indicator variable
tr <- c(rep(1, Nt), rep(0, Nc))
# matrix of included (adjustment) covariates
Z.i <- matrix(runif(K*N), ncol = K)

ret <- mlr.bias.constructor(tr = tr, Z.i = Z.i)

# comparing with brute-force approach
X.i <- cbind(tr, 1, Z.i)
ret2 <- (solve(t(X.i) %*% X.i, t(X.i)))[1, ]

cat("check 1:", all.equal(ret2, ret), "\n")

# sampling with replacement
idx <- sample(1:N, size = round(0.75*N), replace = TRUE)
ret3 <- mlr.bias.constructor(tr = tr, Z.i = Z.i, idx = idx)
ret4 <- (solve(t(X.i[idx, ]) %*% X.i[idx, ], t(X.i[idx, ])))[1, ]

cat("check 2:", all.equal(ret3, ret4), "\n")
```

mlr.combine.bias.variance *Combining bias and variance to produce total MSE for treatment effect*

Description

Combining normalized bias and variance over a range of values for omitted R-squared to produce normalized MSE.

Usage

```
mlr.combine.bias.variance(tr, bvmat, orsq.min = 0.001, orsq.max = 1, n.orsq = 100)
```

Arguments

tr	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
bvmat	Matrix of bias and variances. First column must be bias, and second column must be variance. Each row corresponds to a different 'calibration index' or scenario, which we want to compare and find the best among them.
orsq.min	Minimum omitted R-squared used for combining bias and variance.
orsq.max	Maximum omitted R-squared.
n.orsq	Number of values for omitted R-squared generated in the vector.

Value

A list with the following elements:

orsq.vec	Vector of omitted R-squared values used for combining bias and variance.
errmat	Matrix of MSE, with each row corresponding to an omitted R-squared value, and each column for a value of calibration index, i.e. one row if bvmat.
biassq.mat	Matrix of squared biases, with a structure similar to errmat.
which.min.vec	Value of calibration index (row number for errmat) with minimum MSE.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

`mlr.generate.Z.o` *Generating omitted covariates from included covariates*

Description

Utility function for generating interaction terms and step functions from a set of base covariates, to be used as candidate omitted covariates.

Usage

```
mlr.generate.Z.o(X, interaction.order = 3, step.funcs = TRUE
  , step.thresh = 20, step.ncuts = 3)
```

Arguments

`X`	Matrix of base covariates.
`interaction.order`	Order of interactions to generate. It must be at least 2.
`step.funcs`	Boolean flag, indicating whether (binary) step functions must be generated from continuous variables.
`step.thresh`	Minimum number of distinct values in a numeric vector to generate step functions from.
`step.ncuts`	How many cuts to apply for generating step functions.

Value

TBD

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

`mlr.match` *Thin wrapper around* `Match` *function from* `Matching` *package*

Description

Performs propensity score or Mahalanobis matching and return indexes of treatment and control groups.

Usage

```
mlr.match(tr, X, psm = TRUE, replace = F, caliper = Inf
  , verbose = TRUE)
```

Arguments

`tr`	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
`X`	Covariates used in matching, either directly (Mahalanobis matching) or indirectly (propensity score).
`psm`	Boolean flag, indicating whether propensity score matching should be used (`TRUE`) or Mahalanobis matching (`FALSE`).
`replace`	Boolean flag, indicating whether matching must be done with or without replacement.
`caliper`	Size of caliper (standardized distance of two observations) used in matching. Treatment and control observations with standardized distance larger than `caliper` will not be considered as eligible pairs duing matching.
`verbose`	Boolean flag, indicating whether size of treatment and control groups before and after matching will be printed.

Details

For propensity score matching, linear predictors from logistic regression are used (rather than predicted probabilities).

Value

A vector of matched indexes, containing both treatment and control groups. Also, the following attributes are attached: 1) `nt`: size of treatment group, 2) `nc`: size of control group, 3) `psm.reg`: logistic regression object used in generating propensity scores (`NA` if `psm` is `FALSE`), 4) `match.obj`: matching object returned by `Match` function.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

Examples

```
data(lalonde)

tr <- lalonde$treat
Z.i <- as.matrix(lalonde[, c("age", "educ", "black"
  , "hispan", "married", "nodegree", "re74", "re75")])
Z.o <- model.matrix(~ I(age^2) + I(educ^2) + I(re74^2) + I(re75^2) - 1, lalonde)

# propensity score matching on all covariates
idx <- mlr.match(tr = tr, X = cbind(Z.i, Z.o), caliper = 1.0, replace = FALSE)

# improvement in maximum single-covariate bias due to matching
bias.obj.before <- mlr.bias(tr = tr, Z.i = Z.i, Z.o = Z.o)
bias.before <- bias.obj.before$subspace$bias
dir <- bias.obj.before$subspace$dir
bias.after <- as.numeric(mlr.bias(tr = tr[idx]
  , Z.i = Z.i[idx, ], Z.o = dir[idx], gamma.o = 1.0)$single$bias)

# percentage bias-squared rediction
cat("normalized bias - before:", bias.before, "\n")
cat("normalized bias - after:", bias.after, "\n")
cat("percentage squared-bias reduction:"
  , (bias.before^2 - bias.after^2)/bias.before^2, "\n")

# matching with replacement
idx.wr <- mlr.match(tr = tr, X = cbind(Z.i, Z.o), caliper = 1.0
  , replace = TRUE)
bias.after.wr <- as.numeric(mlr.bias(tr = tr
  , Z.i = Z.i, Z.o = dir, gamma.o = 1.0, idx = idx.wr)$single$bias)
cat("normalized bias - after (with replacement):", bias.after.wr, "\n")
```

`mlr.orthogonalize` *Orthogonalization of vectors with repsect to a matrix*

Description

Decomposing a collection of vectors into parallel and orthogonal components with respect to the subspace spanned by columns of a reference matrix.

Usage

```
mlr.orthogonalize(X, Z, normalize = FALSE, tolerance = .Machine$double.eps^0.5)
```

Arguments

`X`	Matrix whose columns form the subspace, with respect to which we want to orthogonalize columns of `Z`.

Z	Matrix whose columns we want to orthogonalize with respect to the subpsace spanned by columns of X. We must have nrow(Z) == nrow(X).
normalize	Boolean flag, indicating whether the orthogonal component of Z columns must be normalized so that their L2 norms equal nrow(Z) (mean squared is 1).
tolerance	If unnormalized projection of a column of Z has an L2 norm below tolerance, it will not be normalized (even if requested via normalize) and instead a zero vector will be returned.

Details

Current implementation uses Singular Value Decomposition (svd) of X to form an orthonormal basis from columns of X to facilitate the projection process.

Value

A matrix of same dimensions as Z is returned, with each column containing the orthogonal component of the corresponding column of Z. Parallel components are attached as parallel attribute.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

Examples

```
K <- 10
N <- 100
Ko <- 5

X <- matrix(runif(N*K), ncol = K)
Z <- matrix(runif(N*Ko), ncol = Ko)

ret <- mlr.orthogonalize(X = X, Z = Z, normalize = FALSE)

orthogonal <- ret
parallel <- attr(ret, "parallel")
Z.rec <- parallel + orthogonal

# check that parallel and orthogonal components add up to Z
cat("check 1:", all.equal(as.numeric(Z.rec), as.numeric(Z)), "\n")
# check that inner product of orthogonal columns and X columns are zero
cat("check 2:", all.equal(t(orthogonal) %*% X, matrix(0, nrow = Ko, ncol = K)), "\n")
```

`mlr.power` *Power analysis for causal inference using linear regression*

Description

Monte Carlo based calculation of study power for treatment effect estimation using linear regression on treatment indicator and adjustment covariates.

Usage

```
mlr.power(tr, Z.i = NULL, d, sig.level = 0.05, niter = 1000
  , verbose = FALSE, idx = 1:length(tr), rnd = FALSE)
```

Arguments

`tr`	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
`Z.i`	Matrix of adjustment covariates included in linear regression. We must have `nrow(Z.i) == length(tr)`.
`d`	Standardized effect size, equal to treatment effect divided by standard deviation of generative noise.
`sig.level`	Significance level for rejecting null hypothesis.
`niter`	Number of Monte Carlo simulations used for calculating power.
`verbose`	If `TRUE`, calculated power is printed.
`idx`	Subset of observations to use for power calculation.
`rnd`	Boolean flag. If `TRUE`, power is also calculated for random subsampling of observations, using same treatment and control group sizes as indicated by `idx`.

Details

In each Monte Carlo iteration, response variable is generated from a normal distribution whose mean is equal to `d * tr` (other coefficients are assumed to be zero since their value does not affect power calculation), and whose standard deviation is `1.0`. Then OLS-based regression is performed on data, and p-value for treatment effect is compared to `sig.level`, based on which null hypothesis (no effect) is rejected or accepted. The fraction of iterations where null hypothesis is rejected is taken to be power. Standard error is calculated using a binomial-distribution assumption.

Value

A numeric vector is returned. If `rnd` is `FALSE`, meand and standard error of calculated power is returned. If `rnd` is `TRUE`, mean and standard error of power calculated for random subsampling of observations is returned as well.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

mlr.smd *Standardized mean difference*

Description

Calculate standardized mean difference for each column of a matrix, given a binary treatment indicator vector.

Usage

```
mlr.smd(tr, X)
```

Arguments

tr	Binary treatment indicator vector; 1 means treatment, 0 means control.
X	Matrix of covariates; each column is a covariate whose standardized mean difference we want to calculate. nrow(X) must be equal to length(tr).

Value

A vector of length ncol(X), containing standardized mean differences for each column of X, given treatment variable tr.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

mlr.variance *Treatment effect variance*

Description

Calculating treatment effect variance, resulting from linear regression.

Usage

```
mlr.variance(tr, Z.i = NULL, sigsq = 1, details = FALSE
  , idx =1:length(tr))
```

Arguments

tr	Binary treatment indicator vector (1=treatment, 0=control), whose coefficient in the linear regression model is TE.
Z.i	Matrix of adjustment covariates included in linear regression. We must have nrow(Z.i) == length(tr).
sigsq	Variance of data generation noise.
details	Boolean flag, indicating whether intermediate objects used in generating the constrcutor vector must be returned or not (only when no repeated observations).
idx	Index of observations to be used, with possible duplication, e.g. as indexes of matched subset.

Value

A scalar value is returned for TE variance. If details = TRUE and Z.i is not NULL, then the following objects are attached as attributes:

u.i	Vector of length ncol(Z.i), reflecting the mean difference between groups (control - treatment) for each included covariate.
A	Weighted, within-group covariance matrix of included covariates. It is a square matrix of dimension ncol(Z.i).
iA	Inverse of A.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

Examples

```
data(lalonde)

tr <- lalonde$treat
Z.i <- as.matrix(lalonde[, c("age", "educ", "black"
  , "hispan", "married", "nodegree", "re74", "re75")])

ret <- mlr.variance(tr = tr, Z.i = Z.i)

# comparing with brute-force approach
X.i <- cbind(tr, 1, Z.i)
ret2 <- (solve(t(X.i) %*% X.i))[1, 1]

cat("check 1:", all.equal(ret2, ret), "\n")

# matching with/without replacement
idx <- mlr.match(tr = tr, X = Z.i, caliper = 1.0
```

```
  , replace = FALSE)
idx.wr <- mlr.match(tr = tr, X = Z.i, caliper = 1.0
  , replace = TRUE)

ret3 <- mlr.variance(tr = tr, Z.i = Z.i, idx = idx)
cat("variance - matching without replacement:"
  , ret3, "\n")

ret4 <- mlr.variance(tr = tr, Z.i = Z.i, idx = idx.wr)
cat("variance - matching with replacement:"
  , ret4, "\n")
```

`plot.summary.mlr` *Plotting diagnostic and calibration objects resulting from call to* `summary.mlr`

Description

Diagnostic and calibration plots, inlcuding relative squared bias reduction, constrained bias estimation, bias-variance trade-off, and power analysis.

Usage

```
## S3 method for class 'summary.mlr'
plot(x, which = 1
  , smd.index = 1:min(10, ncol(x$smd))
  , bias.index = 1:min(10, ncol(x$bias.terms))
  , orsq.plot = c(0.01, 0.05, 0.25)
  , caption.vec = c("relative squared bias reduction", "normalized bias"
    , "standardized mean difference", "maximum bias"
    , "error components", "optimum choice", "power analysis")
  , ...)
```

Arguments

`x`	An object of class `summary.mlr`, typically the result of a call to `summary.mlr`.
`which`	Selection of which plots to generate: 1 = relative squared bias reduction (by term) for a single idx, 2 = bias terms vs. idx, 3 = standardized mean difference by term vs. idx, 4 = maximum bias (single-covariate, subspace, absolute) vs. idx, 5 = bias/variance/MSE plots, 6 = optimum index vs. omitted r-squared, 7 = power analysis (matched and random subsamples) vs. idx.
`smd.index`	Index of columns in `smd.mat` field of `x` to plot.
`bias.index`	Index of columns in `bias.terms` field of `x` to plot.
`orsq.plot`	Which values for omitted R-squared to generate plots for.
`caption.vec`	Character vector to be used as caption for plots. Values will be repeated if necessary if length is shorter than number of plots requested.
`...`	Parameters to be passed to/from other functions.

Details

Currently, 7 types of plots can be generated, as specified by the `which` flag: 1) relative squared bias reduction, by candidate omitted term, comparing before and after matching, 2) normalized squared bias, by candidate omitted term, vs. calibration index, 3) standardized mean difference, for all included and (candidate) omitted terms, vs. calibration index, 4) aggregate bias (single-covariate maximum, covariate-subspace maximum, and absolute maximum) vs. calibration index, 5) bias/variance/MSE vs. calibration index, at user-supplied values for omitted R-squared, 6) optimal index vs. omitted R-squared, and 7) study power vs. calibration index.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

`summary.mlr` *Applying diagnostic and calibration functions to mlr objects*

Description

Applying a series of diagnostic and calibration functions to a series of matched data sets to determine impact of matching on TE bias, variance and total error, and to select the best matching parameters.

Usage

```
## S3 method for class 'mlr'
summary(object, power = FALSE
  , power.control = list(rnd = TRUE, d = 0.5, sig.level = 0.05
    , niter = 1000, rnd = TRUE)
  , max.method = c("single-covariate", "covariate-subspace"
    , "absolute")
  , verbose = FALSE, ...
  , orsq.min = 1e-03, orsq.max = 1e0, n.orsq = 100)
```

Arguments

`object`	An object of class `mlr`, typically the result of a call to `mlr`.
`power`	Boolean flag indicating whether Monte-Carlo based power analysis must be performed or not.
`power.control`	A list containing parameters to be passed to `mlr.power` for power calculation.
`max.method`	Which constrained bias estimation method must be used in bias-variance trade-off and other analyses?

verbose	Whether progress message must be printed.
...	Parameters to be passed to/from other functions.
orsq.min	Minimum value of omitted R-squared used for combining normalized bias and variance.
orsq.max	Maximum value of omitted R-squared used for combining normalized bias and variance.
n.orsq	Number of values for omitted R-squared to generate in the specified range.

Value

An object of class summary.mlr, with the following elements:

mlr.obj	Same as input.
bias	Matrix of aggregate bias values, one row per calibration index, and three columns: 1) single-covariate maximum, 2) covariate-subspace maximum, and 3) absolute maximum, in that order.
bias.terms	Matrix of biases, one row per calibration index, and one column per candidate omitted term.
variance	Vector of normalized variances, one per each value of calibration index.
power	Matrix of power calculations, one row per calibration index. Each row is identical to output of mlr.power for that calibration index value.
smd	Matrix of standardized mean differences, one row per calibration index, and one column for each included or omitted covariates.
combine.obj	Output of mlr.combine.bias.variance applied to bias and variances at each calibration index value.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Link to a draft paper, documenting the supporting mathematical framework, will be provided in the next release.

Index