

Package ‘matchMulti’

April 18, 2025

Type Package

Title Optimal Multilevel Matching using a Network Algorithm

Version 1.1.14

Date 2025-4-17

Maintainer Sam Pimentel <spi@berkeley.edu>

Description Performs multilevel matches for data with cluster-level treatments and individual-level outcomes using a network optimization algorithm. Functions for checking balance at the cluster and individual levels are also provided, as are methods for permutation-inference-based outcome analysis. Details in Pimentel et al. (2018) <[doi:10.1214/17-AOAS1118](https://doi.org/10.1214/17-AOAS1118)>. The optmatch package, which is useful for running many of the provided functions, may be downloaded from Github at <<https://github.com/markmfredrickson/optmatch>> if not available on CRAN.

Depends R (>= 4.3.0), rlang, dplyr

Imports rebsubset (>= 1.1.4), plyr, coin, weights, mvtnorm, MASS, sandwich, magrittr

Suggests optmatch, testthat, knitr, rrelaxiv

Additional_repositories <https://errickson.net/rrelaxiv/>

License MIT + file LICENSE

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Luke Keele [aut],
Luke Miratrix [aut],
Sam Pimentel [aut, cre],
Paul Rosenbaum [ctb]

Repository CRAN

Date/Publication 2025-04-18 05:50:02 UTC

Contents

balanceMulti	2
buildCaliper	3
catholic_schools	5
describe_data_counts	6
elastic	6
matchMulti	7
matchMultioutcome	11
matchMultiResult	13
matchMultisens	14
minischool	15
rematchSchools	16
tally_schools	18
Index	19

balanceMulti	<i>Performs balance checking after multilevel matching.</i>
--------------	---

Description

This function checks balance after multilevel balance. It checks balance on both level-one (student) and level-two (school) covariates.

Usage

```
balanceMulti(
  match.obj,
  student.cov = NULL,
  school.cov = NULL,
  include.tests = TRUE,
  single.table = FALSE
)
```

Arguments

match.obj	A multilevel match object
student.cov	Names of student level covariates that you want to check balance
school.cov	Names of school level covariates for which you want to check balance, if any.
include.tests	If TRUE include tests for balance. FALSE just report the means and differences.
single.table	If FALSE include a list of student and school covariates separately. TRUE means single balance table.

Details

This function returns a list which include balance checks for before and after matching for both level-one and level-two covariates. Balance statistics include treated and control means, standardized differences, which is the difference in means divided by the pooled standard deviation before matching, and p-values for mean differences. It extracts the matched data and calls ‘balanceTable’ for student and school level covariates.

Value

students	Balance table for student level covariates, as a dataframe.
schools	Balance table for school level covariates, as a dataframe.

Author(s)

Luke Keele, Penn State University, <ljk20@psu.edu> Sam Pimentel, University of Pennsylvania, <spi@wharton.upenn.edu>

See Also

See also [matchMulti](#), [matchMultisens](#), [matchMultioutcome](#), [rematchSchools](#)

Examples

```
## Not run:
# Load Catholic school data
data(catholic_schools)

student.cov <- c('minority','female','ses','mathach')

# Check balance student balance before matching
balanceTable(catholic_schools[c(student.cov,'sector')], treatment = 'sector')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov)

## End(Not run)
```

Description

Fits a propensity score for an individual-level or group-level treatment, computes a caliper for the propensity score (based on a fractional number of standard deviations provided by the user), and creates a matrix containing information about which treated-control pairings are excluded by the caliper.

Usage

```
buildCaliper(data, treatment, ps.vars, group.id = NULL, caliper = 0.2)
```

Arguments

data	A data frame containing the treatment variable, the variables to be used in fitting the propensity score and (if treatment is at the group level) a group ID.
treatment	Name of the treatment indicator.
ps.vars	Vector of names of variables to use in fitting the propensity score.
group.id	Name of group ID variable, if applicable.
caliper	Desired size of caliper, in number of standard deviations of the fitted propensity score.

Details

The treatment variable should be binary with 1 indicating treated units and 0 indicating controls. When `group.id` is `NULL`, treatment is assumed to be at the individual level and the propensity score is fitted using the matrix `data`. When a group ID is specified, data frame `data` is first aggregated into groups, with variables in `ps.vars` replaced by their within-group means, and the propensity score is fitted on the group matrix.

Value

A matrix with `nrow` equal to the number of treated individuals or groups and `ncol` equal to the number of control individuals, with 0 entries indicating pairings permitted by the caliper and `Inf` entries indicating forbidden pairings.

Author(s)

Luke Keele, Penn State University, <ljk20@psu.edu>
Sam Pimentel, University of California, Berkeley, <spi@berkeley.edu>

Examples

```
## Not run:  
# Load Catholic school data  
data(catholic_schools)  
  
student.cov <- c('minority', 'female', 'ses', 'mathach')  
  
# Check balance student balance before matching
```

```

balanceTable(catholic_schools[c(student.cov,'sector')], treatment = 'sector')

#fit a propensity score caliper on mean values of student covariates within schools
school.caliper <- buildCaliper(data = catholic_schools, treatment = 'sector',
ps.vars = student.cov, group.id = 'school')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
school.caliper = school.caliper, school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov)

## End(Not run)

```

catholic_schools *1980 and 1982 High School and Beyond Data*

Description

These data are a subset of the data used in Raudenbush and Bryk (1999) for multilevel modeling.

Format

A data frame with 1595 observations on the following variables.

school: unique school level identifier

ses: student level socio-economic status scale ranges from approx. -3.578 to 2.692

mathach: senior year mathematics test score, outcome measure

female: student level indicator for sex

minority: student level indicator for minority

minority_mean: school level measure of percentage of student body that is minority

female_mean: school level measure of percentage of student body that is female

ses_mean: school level measure of average level of student socio-economic status

sector: treatment indicator 1 if catholic 0 if public

size: school level measure of total number of enrolled students

acad: school level measure of the percentage of students on the academic track

discrm: school level measure of disciplinary climate ranges from approx. -2.4 to 2.7

size_large: school level indicator for schools with more than 1000 students

minority_mean_large: school level indicator for schools with more than ten percent minority

Source

Raudenbush, S. W. and Bryk, A. (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. Thousand Oaks, CA: Sage.

References

United States Department of Education. National Center for Education Statistics. High School and Beyond, 1980: Sophomore and Senior Cohort First Follow-Up (1982).

`describe_data_counts` *Print out summary of student and school counts*

Description

Given a school ID and treatment variable, count up number of schools and students, print out a summary of the counts of students and schools.

Usage

```
describe_data_counts(data, school.id, treatment)
```

Arguments

<code>data</code>	Dataset (student level)
<code>school.id</code>	String name of ID column in data (the grouping variable)
<code>treatment</code>	String name of the treatment variable.

Value

List of three numbers, # control, # Tx, # Total

See Also

`tally_schools`

`elastic` *Optimal Subset Matching without Balance Constraints*

Description

Conducts optimal subset matching as described in the reference.

Usage

```
elastic(mdist, n = 0, val = 0)  
  
pairmatchelastic(mdist, n = 0, val = 0)
```

Arguments

mdist	distance matrix with rows corresponding to treated units and columns corresponding to controls.
n	maximum number of treated units that can be excluded.
val	cost of excluding a treated unit (i.e. we prefer to exclude a treated unit if it increases the total matched distance by more than val).

Details

pairmatchelastic is the main function, which conducts an entire match. elastic is a helper function which augments the original distance matrix as described in the reference.

The original versions of these functions were written by Paul Rosenbaum and distributed in the supplemental material to the paper: "Optimal Matching of an Optimally Chosen Subset in Observational Studies," Paul R. Rosenbaum, Journal of Computational and Graphical Statistics, Vol. 21, Iss. 1, 2012.

Value

elastic returns an augmented version of the input matrix mdist. pairmatchelastic returns a matrix of 1 column whose values are the column numbers of matched controls and whose rownames are the row numbers of matched treated units.

Author(s)

Paul R. Rosenbaum (original forms), modifications by Luke Keele and Sam Pimentel

References

Rosenbaum, Paul R. (2012) "Optimal Matching of an Optimally Chosen Subset in Observational Studies." Journal of Computational and Graphical Statistics, 21.1, 57-71.

matchMulti

A function that performs multilevel matching.

Description

This is the workhorse function in the package which matches groups and units within groups. For example, it will match both schools and students in schools, where the goal is to make units more comparable to estimate treatment effects.

Usage

```

matchMulti(
  data,
  treatment,
  school.id,
  match.students = TRUE,
  student.vars = NULL,
  school.caliper = NULL,
  school.fb = NULL,
  verbose = FALSE,
  keep.target = NULL,
  student.penalty.qtile = 0.05,
  min.keep.pctg = 0.8,
  school.penalty = NULL,
  save.first.stage = TRUE,
  tol = 10,
  solver = "rlemon"
)

```

Arguments

<code>data</code>	A data frame for use in matching.
<code>treatment</code>	Name of covariate that defines treated and control groups.
<code>school.id</code>	Identifier for groups (for example schools)
<code>match.students</code>	Logical flag for whether units within groups should also be matched. If set to FALSE, all units will be retained in both groups.
<code>student.vars</code>	Names of student level covariates on which to measure balance. School-level distances will be penalized when student matches are imbalanced on these variables. In addition, when <code>match.students</code> is TRUE, students are matched on a distance computed from these covariates.
<code>school.caliper</code>	matrix with one row for each treated school and one column for each control school, containing zeroes for pairings allowed by the caliper and Inf values for forbidden pairings. When NULL no caliper is imposed.
<code>school.fb</code>	A list of discrete group-level covariates on which to enforce fine balance, i.e., ensure marginal distributions are balanced. First group is most important, second is second most, etc. If a simple list of variable names, one group is assumed. A list of list will give this hierarchy.
<code>verbose</code>	Logical flag for whether to give detailed output.
<code>keep.target</code>	an optional numeric value specifying the number of treated schools desired in the final match.
<code>student.penalty.qtile</code>	This helps exclude students if they are difficult to match. Default is 0.05, which implies that in the match we would prefer to exclude students rather than match them at distances larger than this quantile of the overall student-student robust Mahalanobis distance distribution

min.keep.pctg	Minimum percentage of students (from smaller school) to keep when matching students in each school pair.
school.penalty	A penalty to remove groups (schools) in the group (school) match
save.first.stage	Should first stage matches be saved.
tol	a numeric tolerance value for comparing distances, used in the school match. It may need to be raised above the default when matching with many levels of refined balance or in very large problems (when these distances will often be at least on the order of the tens of thousands).
solver	Name of package used to solve underlying network flow problem for the school match, one of 'rlemon' and 'rrelaxiv'. rrelaxiv carries an academic license and is not hosted on CRAN so it must be installed separately.

Details

matchMulti first matches students (or other individual units) within each pairwise combination of schools (or other groups); based on these matches a distance matrix is generated for the schools. Then schools are matched on this distance matrix and the student matches for the selected school pairs are combined into a single matched sample.

School covariates are not used to compute the distance matrix for schools (since it is generated from the student match). Instead imbalances in school covariates should be addressed through the `school.fb` argument, which encodes a refined covariate balance constraint. School covariates in `school.fb` should be given in order of priority for balance, since the matching algorithm optimally balances the variables in the first list element, then attempts to further balance the those in the second element, and so on.

Value

raw	The unmatched data before matching.
matched	The matched dataset of both units and groups. Outcome analysis and balance checks are performed on this item.
school.match	Object with two parts. The first lists which treated groups (schools) are matched to which control groups. The second lists the population of groups used in the match.
school.id	Name of school identifier
treatment	Name of treatment variable

Author(s)

Luke Keele, Penn State University, <ljk20@psu.edu>

Sam Pimentel, University of California, Berkeley, <spi@berkeley.edu>

See Also

See also [matchMulti](#), [matchMultisens](#), [balanceMulti](#), [matchMultioutcome](#), [rematchSchools](#)

Examples

```

#toy example with short runtime
library(matchMulti)

#Load Catholic school data
data(catholic_schools)

# Trim data to speed up example
catholic_schools <- catholic_schools[catholic_schools$female_mean >.45 &
  catholic_schools$female_mean < .60,]

#match on a single covariate
student.cov <- c('minority')

  match.simple <-
matchMulti(catholic_schools, treatment = 'sector',
           school.id = 'school', match.students = FALSE,
           student.vars = student.cov, verbose=TRUE, tol=.01)

#Check balance after matching - this checks both student and school balance
  balanceMulti(match.simple, student.cov = student.cov)

## Not run:
#larger example
data(catholic_schools)

student.cov <- c('minority', 'female', 'ses')

# Check balance student balance before matching
balanceTable(catholic_schools[c(student.cov, 'sector')], treatment = 'sector')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
  school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov)

#Estimate treatment effect
output <- matchMultioutcome(match.simple, out.name = "mathach",
  schl_id_name = "school", treat.name = "sector")

# Perform sensitivity analysis using Rosenbaum bound -- increase Gamma to increase effect of
# possible hidden confounder
matchMultisens(match.simple, out.name = "mathach",
  schl_id_name = "school",
  treat.name = "sector", Gamma = 1.3)

# Now match both schools and students within schools

```

```

match.out <- matchMulti(catholic_schools, treatment = 'sector',
  school.id = 'school', match.students = TRUE, student.vars = student.cov)

# Check balance again
bal.tab <- balanceMulti(match.out, student.cov = student.cov)

# Now match with fine balance constraints on whether the school is large
# or has a high percentage of minority students
match.fb <- matchMulti(catholic_schools, treatment = 'sector', school.id = 'school',
  match.students = TRUE, student.vars = student.cov,
  school.fb = list( c('size_large'), c('minority_mean_large') ) )

# Estimate treatment effects
matchMultioutcome(match.fb, out.name = "mathach", schl_id_name = "school", treat.name = "sector")

#Check Balance
balanceMulti(match.fb, student.cov = student.cov)

## End(Not run)

```

matchMultioutcome	<i>Performs an outcome analysis after multilevel matching.</i>
-------------------	--

Description

This function returns a point estimate, 95% confidence interval, and p-values for the matched multilevel data. All results are based on randomization inference.

Usage

```

matchMultioutcome(
  obj,
  out.name = NULL,
  schl_id_name = NULL,
  treat.name = NULL,
  end.1 = -1000,
  end.2 = 1000
)

```

Arguments

obj	A multilevel match object.
out.name	Outcome variable name
schl_id_name	Level 2 ID variable name. This variable identifies the clusters in the data that you want to match.

<code>treat.name</code>	Treatment variable name, must be zero or one.
<code>end.1</code>	Lower bound for point estimate search, default is -1000.
<code>end.2</code>	Upper bound for point estimate search, default is 1000.

Details

It may be necessary to adjust the lower and upper bounds if one expects the treatment effect confidence interval to be outside the range of -1000 or 1000.

Value

<code>pval.c</code>	One-sided approximate p-value for test of the sharp null.
<code>pval.p</code>	One-sided approximate p-value for test of the sharp null assuming treatment effects vary with cluster size
<code>ci1</code>	Lower bound for 95% confidence interval.
<code>ci2</code>	Upper bound for 95% confidence interval.
<code>p.est</code>	Point estimate for the group level treatment effect.

Author(s)

Luke Keele, Penn State University, lj20@psu.edu

Sam Pimentel, University of California, Berkeley, spi@berkeley.edu

References

Rosenbaum, Paul R. (2002) *Observational Studies*. Springer-Verlag.

See Also

See Also as [matchMulti](#), [matchMultisens](#)

Examples

```
## Not run:
# Load Catholic school data
data(catholic_schools)

student.cov <- c('minority', 'female', 'ses', 'mathach')

# Check balance student balance before matching
balanceTable(catholic_schools[c(student.cov, 'sector')], treatment = 'sector')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov)
```

```
#Estimate treatment effect
output <- matchMultioutcome(match.simple, out.name = "mathach",
  schl_id_name = "school", treat.name = "sector")

## End(Not run)
```

matchMultiResult	<i>matchMultiResult</i> object for results of power calculations
------------------	--

Description

The matchMultiResult object is an S3 class that holds the results from the matchMulti call. matchMulti result objects have the matched datasets inside of them.

Usage

```
is.matchMultiResult(x)

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

## S3 method for class 'matchMultiResult'
summary(object, ...)
```

Arguments

x	a matchMultiResult object (except for is.matchMultiResult, where it is a generic object to check).
...	Extra options passed to print.matchMultiResult
object	Object to summarize.

Value

is.matchMultiResult: TRUE if object is a matchMultiResult object.

matchMultisens	<i>Rosenbaum Bounds after Multilevel Matching</i>
----------------	---

Description

Function to calculate Rosenbaum bounds for continuous outcomes after multilevel matching.

Usage

```
matchMultisens(  
  obj,  
  out.name = NULL,  
  schl_id_name = NULL,  
  treat.name = NULL,  
  Gamma = 1  
)
```

Arguments

obj	A multilevel match object
out.name	Outcome variable name
schl_id_name	Level 2 ID variable name, that is this variable identifies clusters matched in the data.
treat.name	Treatment indicator name
Gamma	Sensitivity analysis parameter value. Default is one.

Details

This function returns a single p-value, but actually conducts two tests. The first assumes that the treatment effect does not vary with cluster size. The second allows the treatment effect to vary with cluster size. The function returns a single p-value that is corrected for multiple testing. This p-value is the upper bound for a single Gamma value

Value

pval	Upper bound on one-sided approximate p-value for test of the sharp null.
------	--

Author(s)

Luke Keele, University of Pennsylvania, <luke.keele@gmail.com>

Sam Pimentel, University of California, Berkeley, <spi@berkeley.edu>

References

Rosenbaum, Paul R. (2002) *Observational Studies*. Springer-Verlag.

See Also

See Also as [matchMulti](#), [matchMultioutcome](#)

Examples

```
## Not run:
# Load Catholic school data
data(catholic_schools)

student.cov <- c('minority','female','ses','mathach')

# Check balance student balance before matching
balanceTable(catholic_schools[c(student.cov,'sector')], treatment = 'sector')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov)

#Estimate treatment effect
output <- matchMultioutcome(match.simple, out.name = "mathach",
schl_id_name = "school", treat.name = "sector")

# Perform sensitivity analysis using Rosenbaum bound -- increase Gamma to increase effect of
# possible hidden confounder

matchMultisens(match.simple, out.name = "mathach",
                schl_id_name = "school",
                treat.name = "sector", Gamma=1.3)

## End(Not run)
```

minischool

Mini-data set for illustration

Description

The Catholic schools dataset subset to a smaller number of schools (with only 6 Catholic schools). See full dataset documentation for more information.

Format

A data frame with 1500 rows and 12 variables, as described in the ‘catholic_schools’ dataset.

Source

See documentation page for 'catholic_schools' dataset.

See Also

catholic_schools

rematchSchools	<i>Repeat School Match Only</i>
----------------	---------------------------------

Description

After `matchMulti` has been called, repeats the school match (with possibly different parameters) without repeating the more computationally intensive student match.

Usage

```
rematchSchools(
  match.out,
  students,
  school.fb = NULL,
  verbose = FALSE,
  keep.target = NULL,
  school.penalty = NULL,
  tol = 0.001
)
```

Arguments

<code>match.out</code>	an object returned by a call to <code>matchMulti</code> .
<code>students</code>	a dataframe containing student and school covariates, with a different row for each student.
<code>school.fb</code>	an optional list of character vectors, each containing a subset of the column names of students. Each element of the list should contain all the names in previous elements (producing a nested structure).
<code>verbose</code>	a logical value indicating whether detailed output should be printed.
<code>keep.target</code>	an optional numeric value specifying the number of treated schools desired in the final match.
<code>school.penalty</code>	an optional numeric value, treated as the cost (to the objective function in the underlying optimization problem) of excluding a treated school. If it is set lower, more schools will be excluded.
<code>tol</code>	a numeric tolerance value for comparing distances. It may need to be raised above the default when matching with many levels of refined balance.

Details

The `school.fb` argument encodes a refined covariate balance constraint: the matching algorithm optimally balances the interaction of the variables in the first list element, then attempts to further balance the interaction in the second element, and so on. As such variables should be added in order of priority for balance.

The `keep.target` and `school.penalty` parameters allow optimal subset matching within the school match. When the `keep.target` argument is specified, the school match is repeated for different values of the `school.penalty` parameter in a form of binary search until an optimal match is obtained with the desired number of treated schools or a stopping rule is reached. The `tol` parameter controls the stopping rule; smaller values provide a stronger guarantee of obtaining the exact number of treated schools desired but may lead to greater computational costs.

It is not recommended that users specify the `school.penalty` parameter directly in most cases. Instead the `keep.target` parameter provides an easier way to consider excluding schools.

Author(s)

Luke Keele, Penn State University, <ljk20@psu.edu>

Sam Pimentel, University of California, Berkeley, <spi@berkeley.edu>

References

Rosenbaum, Paul R. (2002). *Observational Studies*. Springer-Verlag.

Rosenbaum, Paul R. (2010). *Design of Observational Studies*. Springer-Verlag.

Rosenbaum, Paul R. (2012) "Optimal Matching of an Optimally Chosen Subset in Observational Studies." *Journal of Computational and Graphical Statistics*, 21.1, 57-71.

See Also

[matchMulti](#).

Examples

```
## Not run:
# Load Catholic school data
data(catholic_schools)

student.cov <- c('minority', 'female', 'ses')
school.cov <- c('minority_mean', 'female_mean', 'ses_mean', 'size', 'acad')

#Match schools but not students within schools
match.simple <- matchMulti(catholic_schools, treatment = 'sector',
  school.id = 'school', match.students = FALSE)

#Check balance after matching - this checks both student and school balance
balanceMulti(match.simple, student.cov = student.cov, school.cov = school.cov)

#now rematch excluding 2 schools
match.trimmed <- rematchSchools(match.simple, catholic_schools, keep.target = 13)
match.trimmed$dropped$schools.t
```

```
## End(Not run)
```

tally_schools	<i>Tally schools and students in a given dataset</i>
---------------	--

Description

Returns a count of schools, without printing anything.

Usage

```
tally_schools(data, school.id, treatment)
```

Arguments

data	Dataset (student level)
school.id	String name of ID column in data (the grouping variable)
treatment	String name of the treatment variable.

Value

List of two things: school and student counts (invisible).

Author(s)

Luke Miratrix

See Also

describe_data_counts

Index

- * **balance**
 - balanceMulti, 2
 - buildCaliper, 3
- * **datasets**
 - catholic_schools, 5
- * **outcomes**
 - matchMultioutcome, 11

- balanceMulti, 2, 9
- buildCaliper, 3

- catholic_schools, 5

- describe_data_counts, 6

- elastic, 6

- is.matchMultiResult (matchMultiResult),
13

- matchMulti, 3, 7, 9, 12, 15, 17
- matchMultioutcome, 3, 9, 11, 15
- matchMultiResult, 13
- matchMultisens, 3, 9, 12, 14
- minischool, 15

- pairmatchelastic (elastic), 6
- print.matchMultiResult
(matchMultiResult), 13

- rematchSchools, 3, 9, 16

- summary.matchMultiResult
(matchMultiResult), 13

- tally_schools, 18