Package 'CMatching'

October 5, 2018

Title Matching Algorithms for Causal Inference with Clustered Data

Version 2.2.1

Date 2018-10-05

Author Massimo Cannas [aut, cre],
Bruno Arpino [ctb],
Elena Colicino [ctb]

Maintainer Massimo Cannas <massimo.cannas@unica.it>

Description
Provides functions to perform matching algorithms for causal inference with clustered data, as described in B. Arpino and M. Cannas (2016) <doi:10.1002/sim.6880>. Pure within-cluster and preferential within-cluster matching are implemented. Both algorithms provide causal estimates with cluster-adjusted estimates of standard errors.

Depends R (>= 2.6.0), Matching

Imports stats,lmtest,multiwayvcov,lme4

Suggests MASS

LazyData false

License GPL-2

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-05 13:32:16 UTC

R topics documented:

CMatching-package *Matching Algorithms for Causal Inference with Clustered Data*

Description

Provides functions to perform matching algorithms for causal inference with clustered data, as described in B. Arpino and M. Cannas (2016) <doi:10.1002/sim.6880>. Pure within-cluster and preferential within-cluster matching are implemented. Both algorithms provide causal estimates with cluster-adjusted estimates of standard errors.

Details

Package:	CMatching
Type:	Package
Version:	2.2
Date:	2018-07-06
License:	GPL version 3 or later

Arpino and Cannas (2016) described several strategies to handle unobserved cluster characteristics in causal inference estimation with clustered data. Depending on researcher's belief about the strength of unobserved cluster level covariates it is possible to take into account clustering either in the estimation of the propensity score model (through the inclusion of fixed or random effects) and/or in the implementation of the matching algorithm. The package contains function `CMatch` to adapt classic matching algorithms for causal inference to clustered data and a customized `summary` function to analyze the output. Depending on the `type` argument function `CMatch` either calls either `MatchW` implementing a *pure* within-cluster matching or function `MatchPW` implementing an approach which can be called *"preferential"* within-cluster matching. The preferential approach first searches for matchable units within the same cluster. If no match was found the algorithm searches in other clusters. The functions also provide causal estimands with cluster-adjusted standard errors from fitting a multilevel model on matched data. `CMatch` returns an object of class "`CMatch`" which can be be summarized and used as input of the `CMatchBalance` function to examine how much the procedure resulted in improved covariate balance. Although `CMatch` has been designed for dealing with clustered data, these algorithms can be used to force a perfect balance or to improve the balance of categorical variables, respectively. In this case, the "clusters" correspond to the levels of the categorical variable(s). When used for this purpouse the user should ignore the standard error (if provided). Note that `Matchby` from package `Matching` can be used for the same purpouse.

Author(s)

Massimo Cannas [aut, cre], Bruno Arpino [ctb], Elena Colicino [ctb] and a special thanks to Thomas W. Yee for his precious help.

Maintainer: Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

Match, MatchBalance

`CMatch` *Within and preferential-within cluster matching.*

Description

This function implements multivariate and propensity score matching in clusters defined by the `Group` variable. It returns an object of class "`CMatch`" which can be be summarized and used as input of the `CMatchBalance` function to examine how much the procedure resulted in improved covariate balance.

Usage

```
CMatch(type, Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
exact = NULL, caliper = 0.25, weights = NULL, replace = TRUE, ties = TRUE, ...)
```

Arguments

`type`	The type of matching desired. "within" for a pure within-cluster matching and "pwithin" for matching preferentially within. The preferential approach first searches for matchable units within the same cluster. If no match was found the algorithm searches in other clusters.
`Y`	A vector containing the outcome of interest.
`Tr`	A vector indicating the treated and control units.
`X`	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.
`Group`	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of `Tr` and `X` containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if `Group` is missing, NULL or it contains only one value the output of the `Match` function is returned with a warning.
`estimand`	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".

M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.
caliper	A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in X are discarded.
weights	A vector of specific observation weights.
replace	Matching can be with or without replacement depending on whether matches can be re-used or not. Default is TRUE.
ties	An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.
...	Additional arguments to be passed to the Match function (not all of them can be used).

Details

This function is meant to be a natural extension of the Match function to clustered data. It retains the main arguments of Match but it has additional output showing matching results cluster by cluster. It differs from wrapper Matchby in package Matching in the way standard errors are calculated and because the caliper is in standard deviation units of the covariates on the overall dataset (so the caliper is the same for all clusters). Moreover, observation weights are available.

Value

index.control	The index of control observations in the matched dataset.
index.treated	The index of control observations in the matched dataset.
index.dropped	The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".
est	The causal estimate. This is provided only if Y is not null. If estimand is "ATT" it is the (weighted) mean of Y in matched treated units minus the (weighted) mean of Y in matched controls. Equivalently, it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.
se	A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: Y ~ constant+Tr, run on the matched dataset (see cluster.vcov for details on how this estimator is obtained). Note that these standard errors differ from a weighted average of cluster specific standard errors provided by the Matchby function, which are generally larger. Estimating standard errors for causal parameters with clustered data is an active field of research and there is no perfect solution to date.

`mdata` A list containing the matched datasets produced by `CMatch`. Three datasets are included in this list: `Y`, `Tr` and `X`. The matched dataset for `Group` can be recovered by `rbind(Group[index.treated],Group[index.control])`.

`orig.treated.nobs.by.group`
The original number of treated observations by group in the dataset.

`orig.control.nobs.by.group`
The original number of control observations by group in the dataset.

`orig.dropped.nobs.by.group`
The number of dropped observations by group after within cluster matching.

`orig.nobs` The original number of observations in the dataset.

`orig.wnobs` The original number of weighted observations in the dataset.

`orig.treated.nobs`
The original number of treated observations in the dataset.

`orig.control.nobs`
The original number of control observations in the dataset.

`wnobs` the number of weighted observations in the matched dataset.

`caliper` The caliper used.

`intcaliper` The internal caliper used.

`exact` The value of the exact argument.

`ndrops.matches` The number of matches dropped either because of the caliper or exact option (or because of forcing the match within-clusters).

`estimand` The estimand required.

Note

The function returns an object of class `CMatch`. The `CMatchBalance` function can be used to examine the covariate balance before and after matching (see the examples below).

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also `Match`, `MatchBalance`

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Suppose that the effect of homeworks on math score is unconfounded conditional on X
# and unobserved school features (we assume this only for illustrative purpouse).

# Let us consider the following variables:

X<-schools$ses # or X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid
# When Group is missing or there is only one Group CMatch returns
# the output of the Match function with a warning.

# Let us assume that the effect of homeworks (Tr) on math score (Y)
# is unconfounded conditional on X and other unobserved school features.
# Several strategies to handle unobserved group characteristics
# are described in Arpino & Cannas, 2016 (see References).


# Multivariate Matching on covariates in X
# default parameters: one-to-one matching on X with replacement with a caliper of 0.25

### Matching within schools
 mw<-CMatch(type="within",Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
 bmw  <- CMatchBalance(Tr~X,data=schools,match.out=mw)

 # calculate proportion of matched observations
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs

 # check number of drops by school
 mw$orig.dropped.nobs.by.group

 # examine output
 mw                   # complete list of results
 summary(mw)  # basic statistics

### Match preferentially within school
# i.e. first match within schools
# then (try to) match remaining units between schools
 mpw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=schools$ses,
 Group=schools$schid, caliper=0.1)

# examine covariate balance
  bmpw<- CMatchBalance(Tr~ses,data=schools,match.out=mpw)
```

```
  # equivalent to MatchBalance(...) with mpw coerced to class "Match"

# proportion of matched observations
  (mpw$orig.treated.nobs-mpw$ndrops) / mpw$orig.treated.nobs
# check drops by school
  mpw$orig.dropped.nobs.by.group.after.pref.within
# proportion of matched observations after match-within only
 (mpw$orig.treated.nobs-sum(mpw$orig.dropped.nobs.by.group.after.within)) / mpw$orig.treated.nobs

# see complete output
   mpw
# or use summary method for main results
   summary(mpw)

#### Propensity score matching

# estimate the ps model

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

# eg 1: within school propensity score matching
psmw <- CMatch(type="within",Y=schools$math, Tr=Tr, X=eps,
Group=schools$schid, caliper=0.1)
# equivalent to direct call at MatchW(Y=schools$math, Tr=Tr, X=eps,
# Group=schools$schid, caliper=0.1)

# eg 2: preferential within school propensity score matching
psmw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# Other strategies for controlling unobserved cluster covariates
# via different specifications of propensity score (see Arpino and Mealli):

# eg 3: propensity score matching using ps estimated from a logit model with dummies for hospitals

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

dpsm <- CMatch(type="within",Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg4: propensity score matching using ps estimated from multilevel logit model
# (random intercept at the hospital level)

require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1 | schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

mpsm<-CMatch(type="within",Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps
```

CMatchBalance *Analyze covariate balance before and after matching.*

Description

Generic function for analyzing covariate balance. If `match.out` is `NULL` only balance statistics for the unmatched data are returned otherwise both before and after matching balance are given. The function is simply a wrapper calling `MatchBalance`, possibly after coercing the class of `match.out`. See `MatchBalance` for more detailed description.

Usage

```
CMatchBalance(match.out, formula, data = NULL, ks = TRUE,
nboots = 500, weights = NULL, digits = 5, paired = TRUE, print.level = 1)
```

Arguments

`match.out`	A matched data set, i.e., the result of a call to `Match` or `CMatch`.
`formula`	This formula does not estimate a model. It is a compact way to describe which variables should be compared between the treated and control group. See `MatchBalance`.
`data`	An optional data set for the variables indicated in the `formula` argument.
`ks`	A flag for whether Kolmogorov-Smirnov tests should be calculated.
`weights`	A vector of observation-specific weights.
`nboots`	The number of bootstrap replication to be used.
`digits`	The number of digits to be displayed in the output
`paired`	A flag for whether a paired t.test should be used for the matched data. An unpaired t.test is always used for unmatched data.
`print.level`	The amount of printing, taking values 0 (no printing), 1(summary) and 2 (dtailed results). Default to 1.

Details

The function is a wrapper of the `MatchBalance` function. If `match.out` is of class `Match` (or `NULL`) then it calls `MatchBalance`. If `match.out` is of class`CMatch` then it coerces the class to `Match` before calling `MatchBalance`. This function is meant to exploit `MatchBalance` for `CMatch` objects for which `MatchBalance` would not work.

Value

Balance statistics for the covariates specified in the *right* side of `formula` argument. Statistics are compared between the two groups specified by the binary variable in the *left* side of `formula`.

Author(s)

Massimo Cannas <massimo.cannas@unica.it> and a special thanks to Thomas W. Yee for his precious help.

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

See Also

MatchBalance

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Suppose that the effect of homeworks on math score is unconfounded conditional on X
# and unobserved school features (we assume this only for illustrative purpouse).

# Let us consider the following variables:

X<-schools$ses # or X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid
# When Group is missing or there is only one Group CMatch returns
# the output of the Match function with a warning.

# Let us assume that the effect of homeworks (Tr) on math score (Y)
# is unconfounded conditional on X and other unobserved school features.
# Several strategies to handle unobserved group characteristics
# are described in Arpino & Cannas, 2016 (see References).


# Multivariate Matching on covariates in X
# default parameters: one-to-one matching on X with replacement with a caliper of 0.25.

### Matching within schools
 mw<-CMatch(type="within",Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
 bmw  <- CMatchBalance(Tr~X,data=schools,match.out=mw)

 # calculate proportion of matched observations
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs
```

```
 # check number of drops by school
 mw$orig.ndrops.by.group


### Match preferentially within school
# i.e. first match within schools
# then (try to) match remaining units between schools
 mpw <- CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=schools$ses,
 Group=schools$schid, caliper=0.1)

# examine covariate balance
  bmpw<- CMatchBalance(Tr~ses,data=schools,match.out=mpw)
  # equivalent to MatchBalance(...) with mpw coerced to class "Match"

# proportion of matched observations
  (mpw$orig.treated.nobs-mpw$ndrops) / mpw$orig.treated.nobs
# check drops by school
  mpw$orig.ndrops.by.group
# proportion of matched observations after match-within only
 (mpw$orig.treated.nobs-sum(mpw$orig.ndrops.by.group.after.within)) / mpw$orig.treated.nobs
```

`MatchPW` *Preferential Within-cluster Matching*

Description

This function implements preferential within-cluster matching. In other words, units that do not match within clusters (as defined by the `Group` variable) can match between cluster in the second step.

Usage

```
MatchPW(Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
 exact = NULL, caliper = 0.25, replace = TRUE, ties = TRUE, weights = NULL, ...)
```

Arguments

`Y`	A vector containing the outcome of interest.
`Tr`	A vector indicating the treated and control units.
`X`	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.
`Group`	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of `Tr` and `X` containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if `Group` is missing, NULL or contains only one value the output of the *Match* function is returned with a warning.

estimand	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".
M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.
caliper	A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in X are discarded. The caliper is used for both within and between clusters matching.
replace	Default is TRUE. Note that setting the parameter to FALSE would give a warning since only the within-matching part can be performed without replacement (see Details).
ties	An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.
weights	A vector of observation specific weights.
...	Please note that all additional arguments of the Match function are not used.

Details

The function performs preferential within-cluster matching in the clusters defined by the variable Group. In the first phase matching within clusters is performed (see MatchW) and in the second the unmatched treated (or controls if estimand="ATC") are matched with all controls (treated) units. This can be helpful to avoid dropping many units in small clusters.

Value

index.control	The index of control observations in the matched dataset.
index.treated	The index of control observations in the matched dataset.
index.dropped	The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".
est	The causal estimate. This is provided only if Y is not null. If estimand is "ATT" it is the (weighted) mean of Y in matched treated minus the (weighted) mean of Y in matched controls. Equivalently it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.
se	A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: y ~ constant+Tr, run on the matched dataset (see cluster.vcov for details on how this estimator is obtained).

`mdata` A list containing the matched datasets produced by `MatchPW`. Three datasets are included in this list: `Y`, `Tr` and `X`. The matched dataset for `Group` can be recovered by `rbind(Group[index.treated],Group[index.control])`.

`orig.treated.nobs.by.group`
The original number of treated observations by group in the dataset.

`orig.control.nobs.by.group`
The original number of control observations by group in the dataset.

`orig.dropped.nobs.by.group`
The number of dropped observations by group after within cluster matching.

`orig.dropped.nobs.by.group.after.pref.within`
The number of dropped observations by group after preferential within group matching.

`orig.nobs` The original number of observations in the dataset.

`orig.wnobs` The original number of weighted observations in the dataset.

`orig.treated.nobs`
The original number of treated observations in the dataset.

`orig.control.nobs`
The original number of control observations in the dataset.

`wnobs` the number of weighted observations in the matched dataset.

`caliper` The caliper used.

`intcaliper` The internal caliper used.

`exact` The value of the exact argument.

`ndrops.matches` The number of matches dropped either because of the caliper or exact option.

`estimand` The estimand required.

Note

The function returns an object of class `CMatch`. The `CMatchBalance` function can be used to examine the covariate balance before and after matching. See the examples below.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also `Match`, `MatchBalance`

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Let us consider the following variables:

X<-schools$ses #X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid
# Note that when Group is missing, NULL or there is only one Group the function
# returns the output of the Match function with a warning.

# Suppose that the effect of homeworks (Tr) on math score (Y)
# is unconfounded conditional on X and other unobserved schools features.
# Several strategies to handle unobserved group characteristics
# are described in Arpino and Cannas, 2016 (see References).


# Multivariate Matching on covariates in X
# default parameters: one-to-one matching on X
# with replacement with a caliper of 0.25; see also \code{Match}).



### Match preferentially within school
# first match within schools
# then (try to) match remaining units between schools
 mpw <- MatchPW(Y=schools$math, Tr=Tr, X=schools$ses, Group=schools$schid, caliper=0.1)
 # equivalent to
 # CMatch(type="pwithin",Y=schools$math, Tr=Tr, X=schools$ses,
 #   Group=schools$schid, caliper=0.1)

# examine covariate balance
  bmpw<- CMatchBalance(Tr~ses,data=schools,match.out=mpw)

# proportion of matched observations
  (mpw$orig.treated.nobs-mpw$ndrops) / mpw$orig.treated.nobs
# check drops by school
  mpw$orig.ndrops.by.group
# proportion of matched observations after match-within only
  (mpw$orig.treated.nobs-sum(mpw$orig.ndrops.by.group.after.within)) / mpw$orig.treated.nobs

# complete output
   mpw
# or use summary method for main results
   summary(mpw)
```

```
#### Propensity score matching

# estimate the propensity score (ps) model

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

# eg 1: preferential within-school propensity score matching
psmw <- MatchPW(Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# We can use other strategies for controlling unobserved cluster covariates
# by using different specifications of ps (see Arpino and Mealli for details):

# eg 2: standard propensity score matching using ps estimated
# from a logit model with dummies for schools

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)



dpsm <- MatchPW(Y=schools$math, Tr=Tr, X=eps, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg3: standard propensity score matching using ps estimated from
# multilevel logit model (random intercept at the school level)


require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1|schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)


mpsm<-MatchPW(Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps
```

MatchW *Within-cluster Matching*

Description

This function implements multivariate and propensity score matching within clusters defined by the Group variable.

Usage

```
MatchW(Y = NULL, Tr, X, Group = NULL, estimand = "ATT", M = 1,
exact = NULL, caliper = 0.25, weights = NULL, replace = TRUE, ties = TRUE, ...)
```

Arguments

Y	A vector containing the outcome of interest.
Tr	A vector indicating the treated and control units.
X	A matrix of covariates we wish to match on. This matrix should contain all confounders or the propensity score or a combination of both.
Group	A vector describing the clustering structure (typically the cluster ID). This can be any numeric vector of the same length of Tr and X containing integer numbers in ascending order otherwise an error message will be returned. Default is NULL, however if Group is missing, NULL or it contains only one value the output of the Match function is returned with a warning.
estimand	The causal estimand desired, one of "ATE", "ATT" and "ATC", which stand for Average Treatment Effect, Average Treatment effect on the Treated and on the Controls, respectively. Default is "ATT".
M	The number of matches which are sought for each unit. Default is 1 ("one-to-one matching").
exact	An indicator for whether exact matching on the variables contained in X is desired. Default is FALSE. This option has precedence over the caliper option.
caliper	A maximum allowed distance for matching units. Units for which no match was found within caliper distance are discarded. Default is 0.25. The caliper is interpreted in standard deviation units of the *unclustered* data for each variable. For example, if caliper=0.25 all matches at distance bigger than 0.25 times the standard deviation for any of the variables in X are discarded.
weights	A vector of specific observation weights.
replace	Matching can be with or without replacement depending on whether matches can be re-used or not. Default is TRUE.
ties	An indicator for dealing with multiple matches. If more than M matches are found for each unit the additional matches are a) wholly retained with equal weights if ties=TRUE; b) a random one is chosen if ties=FALSE. Default is TRUE.
...	Note that additional arguments of the Match function are not used.

Details

This function is meant to be a natural extension of the Match function to clustered data. It retains the main arguments of Match but it has additional output showing matching results cluster by cluster. It differs from wrapper Matchby in package Matching in the way standard errors are calculated and because the caliper is in standard deviation units of the covariates on the overall dataset (so the caliper is the same for all clusters). Moreover, observation weights are available.

Value

`index.control` The index of control observations in the matched dataset.

`index.treated` The index of control observations in the matched dataset.

`index.dropped` The index of dropped observations due to the exact or caliper option. Note that these observations are treated if estimand is "ATT", controls if "ATC".

`est` The causal estimate. This is provided only if `Y` is not null. If estimand is "ATT" it is the (weighted) mean of `Y` in matched treated units minus the (weighted) mean of `Y` in matched controls. Equivalently, it is the weighted average of the within-cluster ATTs, with weights given by cluster sizes in the matched dataset.

`se` A model-based standard error for the causal estimand. This is a cluster robust estimator of the standard error for the linear model: `Y ~ constant+Tr`, run on the matched dataset (see `cluster.vcov` for details on how this estimator is obtained). Note that these standard errors differ from a weighted average of cluster specific standard errors provided by the `Matchby` function, which are generally larger. Estimating standard errors for causal parameters with clustered data is an active field of research and there is no perfect solution to date.

`mdata` A list containing the matched datasets produced by `MatchPW`. Three datasets are included in this list: `Y`, `Tr` and `X`. The matched dataset for `Group` can be recovered by `rbind(Group[index.treated],Group[index.control])`.

`orig.treated.nobs.by.group`
The original number of treated observations by group in the dataset.

`orig.control.nobs.by.group`
The original number of control observations by group in the dataset.

`orig.dropped.nobs.by.group`
The number of dropped observations by group after within cluster matching.

`orig.nobs` The original number of observations in the dataset.

`orig.wnobs` The original number of weighted observations in the dataset.

`orig.treated.nobs`
The original number of treated observations in the dataset.

`orig.control.nobs`
The original number of control observations in the dataset.

`wnobs` the number of weighted observations in the matched dataset.

`caliper` The caliper used.

`intcaliper` The internal caliper used.

`exact` The value of the exact argument.

`ndrops.matches` The number of matches dropped either because of the caliper or exact option (or because of forcing the match within-clusters).

`estimand` The estimand required.

Note

The function returns an object of class `CMatch`. The `CMatchBalance` function can be used to examine the covariate balance before and after matching (see the examples below).

Author(s)

Massimo Cannas <massimo.cannas@unica.it> and E.Colicino

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also Match, MatchBalance

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Let us consider the following variables:

X<-schools$ses #X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid

# Note that when Group is missing, NULL or there is only one group the function returns
# the output of the Match function with a warning.

# Suppose that the effect of homeworks (Tr) on math score (Y)
# is unconfounded conditional on X and other unobserved schools features.
# Several strategies to handle unobserved group characteristics
# are described in Arpino and Cannas, 2016 (see References).


# Multivariate Matching on covariates in X
# default parameters: one-to-one matching on X
# with replacement with a caliper of 0.25; see also \code{Match}.

### Matching within schools
 mw<-MatchW(Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)
 # equivalent to CMatch(type="within",Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
 bmw  <- CMatchBalance(Tr~X,data=schools,match.out=mw)

 # proportion of matched observations
```

```
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs

 # check number of drops by school
 mw$orig.ndrops.by.group

 # examine output
 mw                  # complete results
 summary(mw)         # basic statistics


#### Propensity score matching

# estimate the propensity score (ps) model

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)

# eg 1: within-school propensity score matching
psmw <- MatchW(Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# We can use other strategies for controlling unobserved cluster covariates
# by using different specifications of ps:

# eg 2: standard propensity score matching using ps estimated
# from a logit model with dummies for schools

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)



dpsm <- MatchW(Y=schools$math, Tr=Tr, X=eps, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg3: standard propensity score matching using ps estimated from
# multilevel logit model (random intercept at the school level)

require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1|schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

mpsm<-MatchW(Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps
```

---

schools *Schools data set (NELS-88)*

Description

Data set used by Kreft and De Leeuw in their book *Introducing Multilevel Modeling, Sage (1988)* to analyse the relationship between math score and time spent by students to do math homework. The data set is a subsample of NELS-88 data consisting of 10 handpicked schools from the 1003 schools in the full data set. Students are nested within schools and information is available both at the school and student level.

Usage

```
data("schools")
```

Format

A data frame with 260 observations on the following 19 variables.

`schid` School ID: a numeric vector identyfing each school.

`stuid` The student ID.

`ses` Socioeconomic status.

`meanses` Mean ses for the school.

`homework` The number of hours spent weekly doing homeworks.

`white` A dummy for white race (=1) versus non-white (=0).

`parented` Parents highest education level.

`public` Public school: 1=public, 0=non public.

`ratio` Student-teacher ratio.

`percmin` Percent minority in school.

`math` Math score

`sex` Sex: 1=male, 2=female.

`race` Race of student, 1=asian, 2=Hispanic, 3=Black, 4=White, 5=Native American.

`sctype` Type of school: 1=public, 2=catholic, 3= Private other religion, 4=Private non-r.

`cstr` Classroom environment structure: ordinal from 1=not accurate to 5=very much accurate.

`scsize` School size: ordinal from 1=[1,199) to 7=[1200+).

`urban` Urbanicity: 1=Urban, 2=Suburban, 3=Rural.

`region` Geographic region of the school: NE=1,NC=2,South=3,West=4.

`schnum` Standardized school ID.

Source

Ita G G Kreft, Jan De Leeuw 1988. Introducing Multilevel Modeling, Sage

National Education Longitudinal Study of 1988 (NELS:88): https://nces.ed.gov/surveys/nels88/

Examples

```
data(schools)

# Kreft and De Leeuw, Introducing Multilevel Modeling, Sage (1988).
# The data set is the subsample of NELS-88 data consisting of 10 handpicked schools
# from the 1003 schools in the full data set.

# Suppose that the effect of homeworks on math score is unconfounded conditional on X and
# unobserved school features (we assume this only for illustrative purpouse)

# Let us consider the following variables:

X<-schools$ses #X<-as.matrix(schools[,c("ses","white","public")])
Y<-schools$math
Tr<-ifelse(schools$homework>1,1,0)
Group<-schools$schid
# Note that when Group is missing, NULL or there is only one Group the function
# returns the output of the Match function with a warning.

# Let us assume that the effect of homeworks (Tr) on math score (Y)
# is unconfounded conditional on X and other unobserved schools features.
# Several strategies to handle unobserved group characteristics
# are described in Arpino & Cannas, 2016 (see References).


# Multivariate Matching on covariates in X
#(default parameters: one-to-one matching on X with replacement with a caliper of 0.25).

### Matching within schools
 mw<-MatchW(Y=Y, Tr=Tr, X=X, Group=Group, caliper=0.1)

 # compare balance before and after matching
 bmw  <- MatchBalance(Tr~X,data=schools,match.out=mw)

 # calculate proportion of matched observations
  (mw$orig.treated.nobs-mw$ndrops)/mw$orig.treated.nobs

 # check number of drops by school
 mw$orig.ndrops.by.group

 # examine output
 mw                   # complete list of results
 summary(mw)  # basic statistics


#### Propensity score matching

# estimate the propensity score (ps) model

mod <- glm(Tr~ses+parented+public+sex+race+urban,
family=binomial(link="logit"),data=schools)
eps <- fitted(mod)
```

```
# eg 1: within-school propensity score matching
psmw <- MatchW(Y=schools$math, Tr=Tr, X=eps, Group=schools$schid, caliper=0.1)

# We can use other strategies for controlling unobserved cluster covariates
# by using different specifications of ps (see Arpino and Mealli for details):

# eg 2: standard propensity score matching using ps estimated
# from a logit model with dummies for schools

mod <- glm(Tr ~ ses + parented + public + sex + race + urban
+schid - 1,family=binomial(link="logit"),data=schools)
eps <- fitted(mod)



dpsm <- MatchW(Y=schools$math, Tr=Tr, X=eps, caliper=0.1)
# this is equivalent to run Match with X=eps

# eg3: standard propensity score matching using ps estimated from
# multilevel logit model (random intercept at the school level)

require(lme4)
mod<-glmer(Tr ~ ses + parented + public + sex + race + urban + (1|schid),
family=binomial(link="logit"), data=schools)
eps <- fitted(mod)

mpsm<-MatchW(Y=schools$math, Tr=Tr, X=eps, Group=NULL, caliper=0.1)
# this is equivalent to run Match with X=eps
```

`summary.CMatch` *Summarizing output from MatchW and MatchPW*

Description

summary method for `MatchW` and `MatchPW`

Usage

```
## S3 method for class 'CMatch'
summary(object, ..., full = FALSE, digits = 5)
```

Arguments

`object`	An object of class "`CMatch`".
`...`	Other options for the generic summary function.
`full`	A flag for whether the unadjusted estimates and naive standard errors should also be summarized.
`digits`	The number of significant digits that should be displayed.

Details

A summary of most important output from a "CMatch" object, including size of matched dataset and estimates (if Y is not NULL). If *Group* contains only one value the output is the same of the summary method of package Matching. Otherwise the output shows also the distribution of treated (control) observations *by group* and the distribution of dropped (because of 'caliper' or 'exact' option), also by group.

Note

Naive standard errors are not available when there is more than one group so the full parameter is ineffective in that case.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Sekhon, Jasjeet S. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization. *Journal of Statistical Software 42(7): 1-52.* http://www.jstatsoft.org/v42/i07/

Arpino, B., and Cannas, M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score. *Statistics in Medicine, 35: 2074–2091*. doi: 10.1002/sim.6880.

See Also

See also Match, MatchW, MatchPW, MatchBalance

Index