Package: rsmatch (via r-universe)

September 17, 2024

Title Matching Methods for Time-Varying Observational Studies

Version 0.2.1

Description Implements popular methods for matching in time-varying observational studies. Matching is difficult in this scenario because participants can be treated at different times which may have an influence on the outcomes. The core methods include: ``Balanced Risk Set Matching'' from Li, Propert, and Rosenbaum (2011) <doi:10.1198/016214501753208573> and ``Propensity Score Matching with Time-Dependent Covariates'' from Lu (2005) <doi:10.1111/j.1541-0420.2005.00356.x>. Some functions use the 'Gurobi' optimization back-end to improve the optimization problem speed; the 'gurobi' R package and associated software can be downloaded from <https://www.gurobi.com> after obtaining a license.

License MIT + file LICENSE

URL https://skent259.github.io/rsmatch/, https://github.com/skent259/rsmatch

BugReports https://github.com/skent259/rsmatch/issues

Depends R (>= 2.10)

Imports dplyr, MASS, Matrix, stats

Suggests gurobi, knitr, nbpMatching, Rglpk, rlang, rmarkdown, survival, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Repository https://skent259.r-universe.dev

RemoteUrl https://github.com/skent259/rsmatch

RemoteRef HEAD

RemoteSha 6aa5b3b7c7109cbcbefa2f46f7f25d7ef0101b4e

Contents

---

brsmatch	*Balanced Risk Set Matching*

Description

Perform balanced risk set matching as described in Li et al. (2001) "Balanced Risk Set Matching". Given a longitudinal data frame with covariate information, along with treatment time, build a MIP problem that matches treated individuals to those that haven't been treated yet (or are never treated) based on minimizing the Mahalanobis distance between covariates. If balancing is desired, the model will try to minimize the imbalance in terms of specified balancing covariates in the final pair output. Each treated individual is matched to one other individual.

Usage

```
brsmatch(
  n_pairs,
  data,
  id = "id",
  time = "time",
  trt_time = "trt_time",
  covariates = NULL,
  balance = TRUE,
  balance_covariates = NULL,
  exact_match = NULL,
  options = list(time_lag = FALSE, verbose = FALSE, optimizer = c("glpk", "gurobi"))
)
```

Arguments

n_pairs	The number of pairs desired from matching.
data	A data.frame or similar containing columns matching the id, time, trt_time arguments, and covariates. This data frame is expected to be in tidy, long format, so that id, trt_time, and other variables may be repeated for different values of time. The data.frame should be unique at id and time.
id	A character specifying the id column name (default 'id').
time	A character specifying the time column name (default 'time').
trt_time	A character specifying the treatment time column name (default 'trt_time').

covariates	A character vector specifying the covariates to use for matching (default NULL). If NULL, this will default to all columns except those named by the id, time, and trt_time arguments.
balance	A logical value indicating whether to include balancing constraints in the matching process.
balance_covariates	
	A character vector specifying the covariates to use for balancing (default NULL). If NULL, this will default to all columns except those named by the id, time, and trt_time arguments.
exact_match	A vector of optional covariates to perform exact matching on. If NULL, no exact matching is done.
options	A list of additional parameters with the following components:

- time_lag A logical value indicating whether the matches should be made on the time period preceding treatment. This can help avoid confounding if treatment happens between two periods.
- verbose A logical value indicating whether to print information to the console during a potentially long matching process.
- optimizer The optimizer to use (default 'glpk'). The option 'gurobi' requires an external license and package, but offers speed improvements.

Details

Note that when using exact matching, the n_pairs are split roughly in proportion to the number of treated subjects in each exact matching group. If you would like to control n_pairs exactly, we suggest manually performing exact matching, for example with split(), and selecting n_pairs for each group interactively.

Value

A data frame containing the pair information. The data frame has columns id, pair_id, and type. id matches the input parameter and will contain all ids from the input data frame. pair_id refers to the id of the computed pairs; NA values indicate unmatched individuals. type indicates whether the individual in the pair is considered as treatment ("trt") or control ("all") in that pair.

Author(s)

Sean Kent

References

Li, Yunfei Paul, Kathleen J Propert, and Paul R Rosenbaum. 2001. "Balanced Risk Set Matching." Journal of the American Statistical Association 96 (455): 870-82. doi:10.1198/016214501753208573

Examples

```
if (requireNamespace("Rglpk", quietly = TRUE)) {
  library(dplyr, quietly = TRUE)
  pairs <- brsmatch(
```

```
    n_pairs = 13,
    data = oasis,
    id = "subject_id",
    time = "visit",
    trt_time = "time_of_ad",
    balance = FALSE
  )

  na.omit(pairs)

  # evaluate the first match
  first_match <- pairs$subject_id[which(pairs$pair_id == 1)]
  oasis %>% dplyr::filter(subject_id %in% first_match)
}
```

---

coxpsmatch   *Propensity Score Matching with Time-Dependent Covariates*

Description

Perform propensity score matching as described in Lu (2005) "Propensity Score Matching with
Time-Dependent Covariates". Given a longitudinal data frame with covariate information, along
with treatment time, match treated individuals to those that haven't been treated yet (or are never
treated) based on time-dependent propensity scores from a Cox proportional hazards model. Each
treated individual is matched to one other individual, unless the number of pairs is specified.

Usage

```
coxpsmatch(
  n_pairs = 10^10,
  data,
  id = "id",
  time = "time",
  trt_time = "trt_time",
  covariates = NULL,
  exact_match = NULL,
  options = list(time_lag = FALSE)
)
```

Arguments

n_pairs	The number of pairs desired from matching.
data	A data.frame or similar containing columns matching the id, time, trt_time arguments, and covariates. This data frame is expected to be in tidy, long format, so that id, trt_time, and other variables may be repeated for different values of time. The data.frame should be unique at id and time.
id	A character specifying the id column name (default 'id').

time	A character specifying the time column name (default `'time'`).
trt_time	A character specifying the treatment time column name (default `'trt_time'`).
covariates	A character vector specifying the covariates to use for matching (default NULL). If NULL, this will default to all columns except those named by the id, time, and trt_time arguments.
exact_match	A vector of optional covariates to perform exact matching on. If NULL, no exact matching is done.
options	A list of additional parameters with the following components: • `time_lag` A logical value indicating whether the matches should be made on the time period preceding treatment. This can help avoid confounding if treatment happens between two periods.

Value

A data frame containing the pair information. The data frame has columns id, pair_id, and type. id matches the input parameter and will contain all ids from the input data frame. pair_id refers to the id of the computed pairs; NA values indicate unmatched individuals. type indicates whether the individual in the pair is considered as treatment ("trt") or control ("all") in that pair.

Author(s)

Mitchell Paukner

References

Lu, Bo. 2005. "Propensity Score Matching with Time-Dependent Covariates." Biometrics 61 (3): 721-28. doi:10.1111/j.15410420.2005.00356.x

Examples

```
if (requireNamespace("survival", quietly = TRUE) &
  requireNamespace("nbpMatching", quietly = TRUE)) {
  library(dplyr, quietly = TRUE)
  pairs <- coxpsmatch(
    n_pairs = 13,
    data = oasis,
    id = "subject_id",
    time = "visit",
    trt_time = "time_of_ad"
  )

  na.omit(pairs)

  # evaluate the first match
  first_match <- pairs$subject_id[which(pairs$pair_id == 1)]
  oasis %>% dplyr::filter(subject_id %in% first_match)
}
```

---

oasis   *Longitudinal MRI data in nondemented and demented older adults.*

Description

A dataset containing baseline and time-varying information relating to Alzheimer's disease (AD) based on the Open Access Series of Imaging Studies (OASIS). This set consists of a longitudinal collection of 51 subjects aged 62 to 92. Each subject was scanned on two or more visits, separated by at least one year for a total of 115 imaging sessions. For each subject, 3 or 4 individual T1-weighted MRI scans obtained in single scan sessions are included.

Usage

```
oasis
```

Format

A data frame with 115 rows and 11 variables:

subject_id unique subject identifier

visit visit order

time_of_ad visit in which a patient first had AD diagnosis

m_f male or female

educ years of education

ses socioeconomic status (-1 for missing)

age age of patient at visit

mr_delay MR delay time (contrast)

e_tiv estimated total intracranial volume

n_wbv normalized whole brain volume

asf atlas scaling factor

Details

The data was originally hosted in this Kaggle repository: https://www.kaggle.com/jboysen/mri-and-alzheimers?select=oasis_longitudinal.csv. It has been harmonized for an example analysis for risk set matching based on a reduced sample including patients who go from mild cognitive impairment (MCI) to AD and those patients with MCI throughout.

Source

https://www.kaggle.com/jboysen/mri-and-alzheimers?select=oasis_longitudinal.csv

Index