Package 'FLAME'

April 15, 2020

Type Package

Title Interpretable Matching for Causal Inference

Version 2.0.0

BugReports https://github.com/vittorioorlandi/FLAME/issues

Description Efficient implementations of the algorithms in the Almost-Matching-Exactly framework for interpretable matching in causal inference. These algorithms match units via a learned, weighted Hamming distance that determines which covariates are more important to match on. For more information and examples, see the Almost-Matching-Exactly website.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, magrittr, mice, glmnet, gmp, rlang, tidyr, xgboost, devtools

RoxygenNote 7.0.2

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Vittorio Orlandi [aut, cre], Sudeepa Roy [aut], Cynthia Rudin [aut], Alexander Volfovsky [aut]

Maintainer Vittorio Orlandi <almost.matching.exactly@gmail.com>

Repository CRAN

Date/Publication 2020-04-15 00:30:06 UTC

R topics documented:

ATE *ATE of a matched dataset*

Description

ATE computes the average treatment effect (ATE) of a matched dataset.

Usage

```
ATE(FLAME_out)
```

Arguments

FLAME_out	An object returned by running FLAME

Details

The ATE is computed as the difference between the weighted treated and the weighted control outcomes in the dataset. A unit's weight is the number of times it was matched.

ATT *ATT of a matched dataset*

Description

ATT computes the average treatment effect on the treated (ATT) of a matched dataset.

Usage

```
ATT(FLAME_out)
```

Arguments

FLAME_out	An object returned by running FLAME

Details

The counterfactual outcome of each treated unit is estimated via the mean outcome of control units in its matched group. This value is then averaged across all treated units to compute the ATT.

CATE *Conditional Average Treatment Effects*

Description

CATE returns the conditional average treatment effects (CATEs) of units.

Usage

```
CATE(units, FLAME_out, multiple = FALSE)
```

Arguments

units	A vector of indices for the units whose CATEs are desired.
FLAME_out	The output of a call to FLAME.
multiple	A logical scalar. If FALSE (default), then CATE will return CATEs of main matched groups (those with matches on the greatest number of covariates). See below for details.

Details

The CATE of a matched group is defined to be the difference between average treated and control outcomes within that matched group. When we refer to the CATE(s) of a unit, we mean the CATE(s) of its matched group(s).

Setting multiple = TRUE will request that CATEs corresponding to all matched groups be returned for each unit – if FLAME was run with replace = TRUE to generate FLAME_out in the first place. Otherwise, if FLAME was run with replace = TRUE, but multiple = FALSE, only the CATE of the main matched groups will be returned. The main matched group of a unit contains the first units it matches with (and therefore those with which it matches on the largest number of covariates). If FLAME was run with replace = FALSE, then the user should only supply multiple = FALSE.

Additionally, if FLAME was run with missing_data = 2 to generate FLAME_out, then CATE will return CATE information for all missing_data_imputations imputations.

Value

If passing a single set of matched data

A list of length length(units). Each entry is a CATE (a numeric scalar) (if multiple = FALSE) or a list of CATEs (if multiple = TRUE). If a unit is not matched, the corresponding CATE will be NULL.

Note that this is the return format also if passing a single set of imputed data.

If passing multiple sets of matched, imputed data

A list of length length(FLAME_out), where each entry has the structure described above, corresponding to that imputed data set.

See Also

FLAME

FLAME *Bit Vectors Implementation of FLAME*

Description

FLAME runs the bit-vectors implementation of the FLAME algorithm.

Usage

```
FLAME(
  data,
  holdout = 0.1,
  C = 0.1,
  treated_column_name = "treated",
  outcome_column_name = "outcome",
  binning_method = "sturges",
  PE_method = "ridge",
  user_PE_fit = NULL,
  user_PE_fit_params = NULL,
  user_PE_predict = NULL,
  user_PE_predict_params = NULL,
  replace = FALSE,
  verbose = 2,
  return_pe = FALSE,
  return_bf = FALSE,
  early_stop_iterations = Inf,
  early_stop_epsilon = 0.25,
  early_stop_control = 0,
  early_stop_treated = 0,
  early_stop_pe = Inf,
  early_stop_bf = 0,
  missing_data = 0,
  missing_holdout = 0,
  missing_data_imputations = 5,
  missing_holdout_imputations = 5,
  impute_with_treatment = TRUE,
  impute_with_outcome = FALSE
)
```

Arguments

data	Data to be matched. Either a data frame or a path to a .csv file to be read into a data frame. If path to a .csv file, all covariates will be assumed to be categorical Treatment must be described by a logical or binary column with name

treated_column_name. Outcome, if supplied, must be either binary continuous (both numeric). If not supplied, matching will be performed but matched group CATEs will not be returned and post-matching, treatment effect estimation will not be possible. All non- outcome or treatment columns will be treated as covariates for matching. If they are factors, they will be assumed to be categorical; if they are numeric, they will be assumed continuous and binned into categories as specified by binning_method. *Any covariates that are not continuous, on which units are to match exactly, must be passed to FLAME as factors*. The input of continuous covariates is not recommended. In addition, if a supplied factor has k levels, they must be: 0, 1, ..., k - 1. This will change in a future update. There is no default for data.

holdout Holdout data to be used to compute predictive error. If a numeric scalar between 0 and 1, that proportion of data will be made into a holdout set and only the remaining proportion of data will be matched. Otherwise, a dataframe or a path to a .csv file. If a path to a .csv file, all covariates will be assumed to be categorical Restrictions on column types are the same as for data. Must have the same column names and order as data. This data will *not* be matched. Defaults to 0.1.

C A finite, positive scalar denoting the tradeoff between BF and PE in the FLAME algorithm. Higher C prioritizes more matches and lower C prioritizes not dropping important covariates. Defaults to 0.1.

treated_column_name
A character with the name of the treatment column in data and holdout. Defaults to 'treated'.

outcome_column_name
A character with the name of the outcome column in holdout and also in data, if supplied in the latter. Defaults to 'outcome'.

binning_method The method to be used to bin continuous covariates in the data. One of: "sturges", "scott", or "fd", denoting Sturges' rule, Scott's rule, or the Freedman-Diaconis rule for determining number of bins in a histogram. Each continuous covariate will be binned into the corresponding number of bins. If covariates are binned, the data entry of the object returned from FLAME will contain the binned, and not the original, values. Defaults to 'sturges'.

PE_method Either "ridge" or "xgb". Denotes the method to be used to compute PE. If "ridge", uses glmnet::cv.glmnet with default parameters and then the default predict method to estimate the outcome. If "xgb", uses xgboost::xgb.cv on a wide range of parameter values to cross-validate and find the best with respect to RMSE (for continuous outcomes) or binary misclassification rate (for binary outcomes). Then uses the default predict method to estimate the outcome. Defaults to "ridge".

user_PE_fit An optional function supplied by the user that can be used instead of those allowed for by PE_method to fit a model fitting the outcome from the covariates. Must take in a matrix of covariates as its first argument and a vector outcome as its second argument. Defaults to NULL.

user_PE_fit_params
A named list of optional parameters to be used by user_PE_fit. Defaults to NULL.

`user_PE_predict`
: An optional function supplied by the user that can be used to generate predictions from the output of `user_PE_fit`. As its first argument, must take an object of the type returned by `user_PE_fit` and as its second, a matrix of values for which to generate predictions. If not supplied, defaults to `predict`.

`user_PE_predict_params`
: A named list of optional parameters to be used by `user_PE_params`. Defaults to `NULL`.

`replace`
: A logical scalar. If `TRUE`, allows the same unit to be matched multiple times, on different sets of covariates. In this case, balancing factor is computing by dividing by the total number of treatment (control) units, instead of the number of unmatched treatment (control) units. Defaults to `FALSE`.

`verbose`
: Controls how FLAME displays progress while running. If 0, no output. If 1, only outputs the stopping condition. If 2, outputs the iteration and number of unmatched units every 5 iterations, and the stopping condition. If 3, outputs the iteration and number of unmatched units every iteration, and the stopping condition. Defaults to 2.

`return_pe`
: A logical scalar. If `TRUE`, the predictive error (PE) at each iteration will be returned. Defaults to `FALSE`.

`return_bf`
: A logical scalar. If `TRUE`, the balancing factor (BF) at each iteration will be returned. Defaults to `FALSE`.

`early_stop_iterations`
: A nonnegative integer, denoting an upper bound on the number of iterations of FLAME to be performed. If 0, one round of exact matching is performed before stopping. Defaults to `Inf`.

`early_stop_epsilon`
: A nonnegative numeric. If FLAME attemts to drop a covariate that would raise the PE above (1 + early_stop_epsilon) times the baseline PE (the PE before any covariates have been dropped), FLAME will stop. Defaults to 0.25.

`early_stop_control`
: A numeric value between 0 and 1. If the proportion of control units that are unmatched falls below this value, FLAME stops. Defaults to 0.

`early_stop_treated`
: A numeric value between 0 and 1. If the proportion of treatment units that are unmatched falls below this value, FLAME stops. Defaults to 0.

`early_stop_pe`
: A numeric value between 0 and 1. If FLAME attempts to drop a covariate that would lead to a PE above this value, FLAME stops. Defaults to `Inf`.

`early_stop_bf`
: A numeric value between 0 and 1. If FLAME attempts to drop a covariate that would lead to a BF below this value, FLAME stops. Defaults to 0.

`missing_data`
: If 0, assumes no missingness in `data`. If 1, does not match units with missingness in `data`. In this case, the balancing factor is computed ignoring units with missingness. If 2, generates `missing_data_imputations` imputed datasets via `mice::mice`. In this case, the results of running `FLAME` on each imputed dataset will be returned in a list. Within each of these list entries, the `data` entry will contain the imputed, not missing, values. If 3, will not match a unit on a covariate that it is missing. Defaults to 0.

`missing_holdout`
: If 0, assumes no missing data in `holdout`. If 1, eliminates units with missingness from `holdout`. If 2, generates `missing_holdout_imputations` imputed datasets via `mice::mice`. In this latter case, all imputations will be used to compute PE, and the PE at an iteration will be the average across all imputations. Defaults to 0.

`missing_data_imputations`
: If `missing_data = 2`, performs this many imputations of the missing data in `data` via `mice::mice`. Defaults to 5.

`missing_holdout_imputations`
: If `missing_holdout = 2`, performs this many imputations of the missing data in `holdout` via `mice::mice`. Defaults to 5.

`impute_with_treatment`
: A logical scalar. If `TRUE`, uses treatment assignment to impute covariates when `missing_data = 2` or `missing_holdout = 2`. Defaults to `TRUE`.

`impute_with_outcome`
: A logical scalar. If `TRUE`, uses outcome information to impute covariates when `missing_data = 2` or `missing_holdout = 2`. Defaults to `FALSE`.

Value

The basic object returned by `FLAME` is a list of 6 entries:

data The original data frame with several modifications:

1. An extra logical column, `data$matched`, that indicates whether or not a unit was matched.
2. An extra numeric column, `data$weight`, that denotes on how many different sets of covariates a unit was matched. This will only be greater than 1 when `replace = TRUE`.
3. Regardless of their original names, the columns denoting treatment and outcome in the data will be renamed 'treated' and 'outcome' and they are moved to be located after all the covariate data.
4. Units that were not matched on all covariates will have a * in place of their covariate value for all covariates on which they were not matched.

MGs A list of all the matched groups formed by FLAME. Each entry contains the units in a single matched group

CATE A numeric vector with the conditional average treatment effect of every matched group in `MGs`

matched_on A list corresponding to `MGs` that gives the covariates, and their values, on which units in each matched group were matched.

matching_covs A list with the covariates used for matching on every iteration of FLAME

dropped A vector with the covariate dropped at each iteration of FLAME

Introduction

FLAME is a matching algorithm for causal inference that matches units if they match exactly on certain covariates. It starts by making any possible matches on all covariates. It then drops a covariate, makes any possible matches on the remaining covariates, and repeats this process until stopping. The covariate dropped at any given iteration is the one yielding the greatest match quality

MQ, defined as $MQ = C \times BF - PE$. Here, BF denotes the balancing factor, defined as the proportion of unmatched control units, plus the proportion of unmatched treated units, that can now be matched by dropping that covariate. And PE denotes the prediction error, defined as the training error incurred when predicting the outcome from covariates on a separate, holdout set. In this way, FLAME encourages making many matches and also matching on covariates important to the outcome. The hyperparameter C controls the balance between these two objectives. For more details, please see the FLAME paper here.

Stopping Rules

By default, `FLAME` stops when 1. all covariates have been dropped or 2. all treatment or control units have been matched. This behavior can be modified by the arguments whose prefix is "early_stop". With the exception of `early_stop_iterations`, all the rules come into play *before* the offending covariate is dropped. That is, if `early_stop_BF = 0.2` and at the current iteration, dropping the covariate leading to highest match quality is associated with a balancing factor of 0.1, FLAME stops *without* dropping this covariate.

Missing Data

`FLAME` offers functionality for handling missing data in the covariates, for both the `data` and `holdout` sets. This functionality can be specified via the arguments whose prefix is "missing" or "impute". It allows for ignoring missing data, imputing it, or (for `data`) not matching on missing values. If `data` is imputed, the FLAME algorithm will be run on all imputations. If `holdout` is imputed, the predictive error at an iteration will be the average of predictive errors across all imputed `holdout` datasets.

Examples

```
data <- gen_data()
holdout <- gen_data()
FLAME_out <- FLAME(data = data, holdout = holdout)
```

`gen_data` *Generate Toy Data for Matching*

Description

`gen_data` generates toy data that can be used to explore FLAME's functionality.

Usage

```
gen_data(n = 250, p = 5, write = FALSE, path = getwd(), filename = "FLAME.csv")
```

Arguments

n	Number of units desired in the data set. Defaults to 250.
p	Number of covariates in the data set. Must be greater than 2. Defaults to 5.
write	A logical scalar. If TRUE, the resulting data is stored as a .csv file as specified by arguments path and filename. Defaults to FALSE.
path	The path to the location where the data should be written if write = TRUE. Defaults to getwd().
filename	The name of the file to which the data should be written if write = TRUE. Defaults to FLAME.csv.

Details

gen_data simulates data in the format accepted by FLAME. Covariates X_i and treatment T are both independently generated according to a Bernoulli(0.5) distribution. The outcome Y is generated according to $Y = 15X_1 - 10X_2 + 5X_3 + 5T + \epsilon$, where $\epsilon \sim N(0, I_n)$. Thus, the value of p must be at least 3 and any additional covariates beyond X_1, X_2, X_3 are irrelevant.

Value

A data frame that may be passed to FLAME. Covariates are categorical and therefore coded as factors. Treatment is binary numeric and outcome is numeric.

MG *Matched Groups*

Description

MG returns the matched groups of the supplied units.

Usage

```
MG(units, FLAME_out, multiple = FALSE, index_only = FALSE)
```

Arguments

units	A vector of indices for the units whose matched groups are desired.
FLAME_out	The output of a call to FLAME.
multiple	A logical scalar. If FALSE (default), then MG will only return a main matched group for each unit (the first matched group that unit was a part of). See below for details.
index_only	A logical scalar. If TRUE then only the indices of units in each matched group are returned.

Details

By default, MG returns the covariate, treatment, and outcome information for all the units in the relevant matched groups. If only the indices of units in the matched groups are desired, index_only can be set to TRUE.

Setting multiple = TRUE will request that all matched groups be returned for each unit – if FLAME was run with replace = TRUE to generate FLAME_out in the first place. Otherwise, if FLAME was run with replace = TRUE, but multiple = FALSE, only main matched groups will be returned. The main matched group of a unit contains the first units it matches with (and therefore those with which it matches on the largest number of covariates). If FLAME was run with replace = FALSE, then the user should only supply multiple = FALSE.

Additionally, if FLAME was run with missing_data = 2 to generate FLAME_out, then MG will return matched group information for all missing_data_imputations imputations.

Value

If passing a single set of matched data

A list of length length(units). Each entry is a data frame (if multiple = FALSE) or a list of data frames (if multiple = TRUE). For a given entry, these data frames are subsets of data passed to FLAME to generate FLAME_out, whose rows correspond to the units in the matched group(s) of that entry. If a unit is not matched, the corresponding CATE will be NULL.

The starred entries (*) in the returned data frames have the same meaning as in FLAME_out$data, except for if both multiple = TRUE and replace = TRUE. In this case, if *all* units do not match on a given covariate, all entries of that covariate will be starred, even though a subset of the units may have matched on them. This is done so that it is clear on which covariates these units match.

Note that this is the return format also if passing a single set of imputed data.

If passing multiple sets of matched, imputed data

A list of length length(FLAME_out), where each entry has the structure described above, corresponding to that imputed data set.

See Also

FLAME

Index