Package 'haldensify'

March 14, 2020

Title Highly Adaptive Lasso Conditional Density Estimation

Version 0.0.5

Maintainer Nima Hejazi <nh@nimahejazi.org>

Description Conditional density estimation is a longstanding and challenging problem in statistical theory, and numerous proposals exist for optimally estimating such complex functions. Algorithms for nonparametric estimation of conditional densities based on a pooled hazard regression formulation and semiparametric estimation via conditional hazards modeling are implemented based on the highly adaptive lasso, a nonparametric regression function for efficient estimation with fast convergence under mild assumptions. The pooled hazards formulation implemented was first described by Díaz and van der Laan (2011) <doi:10.2202/1557-4679.1356>.

Depends R (>= 3.2.0)

Imports stats, ggplot2, data.table, future.apply, assertthat, hal9001 (>= 0.2.5), origami (>= 1.0.0), Rdpack

Suggests testthat, knitr, rmarkdown, future, dplyr

License MIT + file LICENSE

URL https://github.com/nhejazi/haldensify

BugReports https://github.com/nhejazi/haldensify/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

RdMacros Rdpack

NeedsCompilation no

Author Nima Hejazi [aut, cre, cph] (<https://orcid.org/0000-0002-7127-2789>), David Benkeser [aut] (<https://orcid.org/0000-0002-1019-8343>), Mark van der Laan [aut, ths] (<https://orcid.org/0000-0003-1432-5511>)

Repository CRAN

Date/Publication 2020-03-14 15:20:05 UTC

R topics documented:

`cv_haldensify` *Conditional density estimation with HAL in a single cross-validation fold*

Description

Conditional density estimation with HAL in a single cross-validation fold

Usage

```
cv_haldensify(
  fold,
  long_data,
  wts = rep(1, nrow(long_data)),
  lambda_seq = exp(seq(-1, -13, length = 100))
)
```

Arguments

`fold`	Object specifying cross-validation folds as generated by a call to `make_folds`.
`long_data`	A `data.table` or `data.frame` object containing the data in long format, as given in Díaz I, van der Laan MJ (2011). "Super learner based conditional density estimation with application to marginal structural models." *The International Journal of Biostatistics*, **7**(1), 1–20., as produced by `format_long_hazards`.
`wts`	A `numeric` vector of observation-level weights, matching in its length the number of records present in the long format data. Default is to weight all observations equally.
`lambda_seq`	A `numeric` sequence of values of the tuning parameter of the Lasso L1 regression passed to `fit_hal`.

Details

Estimates the conditional density of A|W for a subset of the full set of observations based on the inputted structure of the cross-validation folds. This is a helper function intended to be used to select the optimal value of the penalization parameter for the highly adaptive lasso estimates of the conditional hazard (via `cross_validate`). The

Value

A list, containing density predictions, observations IDs, observation-level weights, and cross-validation indices for conditional density estimation on a single fold of the overall data.

---

format_long_hazards *Generate long format hazards data for conditional density estimation*

Description

Generate long format hazards data for conditional density estimation

Usage

```
format_long_hazards(
  A,
  W,
  wts = rep(1, length(A)),
  grid_type = c("equal_range", "equal_mass"),
  n_bins = NULL,
  breaks = NULL
)
```

Arguments

A	The numeric vector or similar of the observed values of an intervention for a group of observational units of interest.
W	A data.frame, matrix, or similar giving the values of baseline covariates (potential confounders) for the observed units whose observed intervention values are provided in the previous argument.
wts	A numeric vector of observation-level weights. The default is to weight all observations equally.
grid_type	A character indicating the strategy (or strategies) to be used in creating bins along the observed support of the intervention A. For bins of equal range, use "equal_range"; consult documentation of cut_interval for more information. To ensure each bin has the same number of points, use "equal_mass"; consult documentation of cut_number for details.
n_bins	Only used if grid_type is set to "equal_range" or "equal_mass". This numeric value indicates the number(s) of bins into which the support of A is to be divided.
breaks	A numeric vector of break points to be used in dividing up the support of A. This is passed through the ... argument to cut.default by cut_interval or cut_number.

Details

Generates a long-form dataset that represents each observation in terms of repeated measures across discretized bins derived from selecting break points over the support of A. This repeated measures dataset is suitable for estimating the hazard of failing in a particular bin over A using a highly adaptive lasso classification model.

Value

A `list` containing the break points used in dividing the support of `A` into discrete bins, the length of each bin, and the reformatted data. The reformatted data is a `data.table` of repeated measures data, with an indicator for which bin an observation fails in, the bin ID, observation ID, values of `W` for each given observation, and observation-level weights.

`haldensify` *Cross-validated conditional density estimation with HAL*

Description

Cross-validated conditional density estimation with HAL

Usage

```
haldensify(
  A,
  W,
  wts = rep(1, length(A)),
  grid_type = c("equal_range", "equal_mass"),
  n_bins = c(5, 10),
  lambda_seq = exp(seq(-1, -13, length = 1000)),
  use_future = FALSE,
  seed_int = 791L
)
```

Arguments

`A`	The `numeric` vector or similar of the observed values of an intervention for a group of observational units of interest.
`W`	A `data.frame`, `matrix`, or similar giving the values of baseline covariates (potential confounders) for the observed units whose observed intervention values are provided in the previous argument.
`wts`	A `numeric` vector of observation-level weights. The default is to weight all observations equally.
`grid_type`	A `character` indicating the strategy (or strategies) to be used in creating bins along the observed support of the intervention `A`. For bins of equal range, use "equal_range"; consult documentation of `cut_interval` for more information. To ensure each bin has the same number of points, use "equal_mass"; consult documentation of `cut_number` for details.

n_bins	Only used if type is set to "equal_range" or "equal_mass". This numeric value indicates the number(s) of bins into which the support of the intervention A is to be divided.
lambda_seq	A numeric sequence of values of the tuning parameter of the Lasso L1 regression passed to fit_hal.
use_future	A logical indicating whether to attempt to use parallelization based on the **future** and **future.apply** packages. If set to TRUE, future_mapply will be used in place of mapply. When set to TRUE, a parallelization scheme must be set externally by using plan.
seed_int	An integer used to set the seed in the cross-validation procedure used to select binning values. This is passed to the argument future.seed of future_mapply.

Details

Estimation of the conditional density A|W through using the highly adaptive lasso to estimate the conditional hazard of failure in a given bin over the support of A. Cross-validation is used to select the optimal value of the penalization parameters, based on minimization of the weighted log-likelihood loss for a density.

Value

Object of class haldensify, containing a fitted hal9001 object, a vector of break points used in binning A over its support W, sizes of the bins used in each fit, the tuning parameters selected by cross-validation, and the range of the A.

Examples

```
# simulate data: W ~ U[-4, 4] and A|W ~ N(mu = W, sd = 0.5)
n_train <- 50
w <- runif(n_train, -4, 4)
a <- rnorm(n_train, w, 0.5)
# learn relationship A|W using HAL-based density estimation procedure
mod_haldensify <- haldensify(
  A = a, W = w, n_bins = 3,
  lambda_seq = exp(seq(-1, -10, length = 50))
)
```

---

map_hazard_to_density *Map a predicted hazard to a predicted density for a single observation*

Description

Map a predicted hazard to a predicted density for a single observation

Usage

```
map_hazard_to_density(hazard_pred_single_obs)
```

Arguments

`hazard_pred_single_obs`
: A `numeric` vector of predicted hazard of failure in a given bin (under a given partitioning of the support) for a single observational unit based on a long format data structure (from `format_long_hazards`). This is simply the probability that the observed value falls in a corresponding bin, given that it has not yet failed (fallen in a previous bin), as described in Díaz I, van der Laan MJ (2011). "Super learner based conditional density estimation with application to marginal structural models." *The International Journal of Biostatistics*, **7**(1), 1–20..

Details

For a single observation, map a predicted hazard of failure (as an occurrence in a particular bin, under a given partitioning of the support) to a density.

Value

A `matrix` composed of a single row and a number of columns specified by the grid of penalization parameters used in fitting of the highly adaptive lasso. This is the predicted conditional density for a given observation, re-mapped from the hazard scale.

`predict.haldensify` *Prediction method for HAL-based conditional density estimation*

Description

Prediction method for HAL-based conditional density estimation

Usage

```
## S3 method for class 'haldensify'
predict(object, ..., new_A, new_W)
```

Arguments

`object`
: An object of class `haldensify`, containing the results of fitting the highly adaptive lasso for conditional density estimation, as produced by a call to `haldensify`.

`...`
: Additional arguments passed to `predict` as necessary.

`new_A`
: The `numeric` vector or similar of the observed values of an intervention for a group of observational units of interest.

`new_W`
: A `data.frame`, `matrix`, or similar giving the values of baseline covariates (potential confounders) for the observed units whose observed intervention values are provided in the previous argument.

Details

Method for computing and extracting predictions of the conditional density estimates based on the highly adaptive lasso estimator, returned as an S3 object of class `haldensify` from `haldensify`.

Value

A `numeric` vector of predicted conditional density values from a fitted `haldensify` object.

Examples

```
# simulate data: W ~ U[-4, 4] and A|W ~ N(mu = W, sd = 0.5)
n_train <- 50
w <- runif(n_train, -4, 4)
a <- rnorm(n_train, w, 0.5)
# learn relationship A|W using HAL-based density estimation procedure
mod_haldensify <- haldensify(
  A = a, W = w, n_bins = 3,
  lambda_seq = exp(seq(-1, -10, length = 50))
)
# predictions to recover conditional density of A|W
new_a <- seq(-4, 4, by = 0.1)
new_w <- rep(0, length(new_a))
pred_dens <- predict(mod_haldensify, new_A = new_a, new_W = new_w)
```

Index