Package ‘cforward’

October 12, 2022

Title  Forward Selection using Concordance/C-Index
Version  0.1.0
Description  Performs forward model selection, using the C-index/concordance
             in survival analysis models.
License  GPL-3
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.1
Imports  survival, dplyr, stats, magrittr, tibble
URL  https://github.com/muschellij2/cforward
BugReports  https://github.com/muschellij2/cforward/issues
Depends  R (>= 2.10)
Suggests  testthat
NeedsCompilation  no
Author  John Muschelli [aut, cre] (<https://orcid.org/0000-0001-6469-1750>),
        Andrew Leroux [aut]
Maintainer  John Muschelli <muschellij2@gmail.com>
Repository  CRAN
Date/Publication  2021-03-29 14:20:08 UTC

R topics documented:
cforward ......................................................... 2
estimate_concordance ................................. 4
nhanes_example ................................. 5

Index  6
**Description**

Forward Selection Based on C-Index/Concordance

**Usage**

```r
cforward(
  data,
  event_time = "event_time_years",
  event_status = "mortstat",
  weight_column = "WTMEC4YR_norm",
  variables = NULL,
  included_variables = NULL,
  n_folds = 10,
  seed = 1989,
  max_model_size = 50,
  c_threshold = NULL,
  verbose = TRUE,
  cfit_args = list(),
  save_memory = FALSE,
  ...
)

cforward_one(
  data,
  event_time = "event_time_years",
  event_status = "mortstat",
  weight_column = "WTMEC4YR_norm",
  variables,
  included_variables = NULL,
  verbose = TRUE,
  cfit_args = list(),
  save_memory = FALSE,
  ...
)

make_folds(data[, event_status = "mortstat", n_folds = 10, verbose = TRUE])
```

**Arguments**

- **data**
  A data set to perform model selection and cross-validation.

- **event_time**
  Character vector of length 1 with event times, passed to `Surv`

- **event_status**
  Character vector of length 1 with event status, passed to `Surv`
weight_column  Character vector of length 1 with weights for model. If no weights are available, set to NULL.
variables     Character vector of variables to perform selection. Must be in data.
included_variables  Character vector of variables forced to have in the model. Must be in data.
n_folds       Number of folds for Cross-validation. If you want to run on the full data, set to 1.
seed          Seed set before folds are created.
max_model_size maximum number of variables in the model. Selection will stop if reached. Note, this does not correspond to the number of coefficients, due to categorical variables.
c_threshold   threshold for concordance. If the difference in the best concordance and this one does not reach a certain threshold, break.
verbose       print diagnostic messages

Value
A list of lists, with elements of:

full_concordance  Concordance when fit on the full data
models  Cox model from full data set fit, stripped of large memory elements
cv_concordance Cross-validated Concordance
included_variables  Variables included in the model, other than those being selection upon

Examples
variables = c("gender",
              "age_years_interview", "education_adult")
res = cforward(nhanes_example,
               event_time = "event_time_years",
               event_status = "mortstat",
               weight_column = "WTMEC4YR_norm",
               variables = variables,
               included_variables = NULL,
               n_folds = 5,
               c_threshold = 0.02,
               seed = 1989,
               max_model_size = 50,
               verbose = TRUE)
conc = sapply(res, `[`, "best_concordance")
```r
res = cforward(nhanes_example,
    event_time = "event_time_years",
    event_status = "mortstat",
    weight_column = "WTMEC4YR_norm",
    variables = variables,
    included_variables = NULL,
    n_folds = 5,
    seed = 1989,
    max_model_size = 50,
    verbose = TRUE)
conc = sapply(res, 
  function(x) x[[1]], "best_concordance")
threshold = 0.01
included_variables = names(conc)[c(1, diff(conc)) > threshold]
new_variables = c("diabetes", "stroke")
second_level = cforward(nhanes_example,
    event_time = "event_time_years",
    event_status = "mortstat",
    weight_column = "WTMEC4YR_norm",
    variables = new_variables,
    included_variables = included_variables,
    n_folds = 5,
    seed = 1989,
    max_model_size = 50,
    verbose = TRUE)
second_conc = sapply(second_level, 
  function(x) x[[1]], "best_concordance")
result = second_level[[which.max(second_conc)]]
final_model = result$models[[which.max(result$cv_concordance)]]
```

---

**estimate_concordance**  
*Estimate Out-of-Sample Concordance*

**Description**

Estimate Out-of-Sample Concordance

**Usage**

```r
estimate_concordance(
    train,
    test = train,
    event_time = "event_time_years",
    event_status = "mortstat",
    weight_column = "WTMEC4YR_norm",
    all_variables = NULL,
    cfit_args = list(),
    ...
)
```
Arguments

- **train**: A data set to perform model training.
- **test**: A data set to estimate concordance, from fit model with `train`. Set to `train` if estimating on the same data.
- **event_time**: Character vector of length 1 with event times, passed to `Surv`.
- **event_status**: Character vector of length 1 with event status, passed to `Surv`.
- **weight_column**: Character vector of length 1 with weights for model. If no weights are available, set to `NULL`.
- **all_variables**: Character vector of variables to put in the model. All must be in `data`.
- **cfit_args**: Arguments passed to `concordancefit`. If `strata` is to be passed, set `strata_column` in this list.
- **...**: Additional arguments to pass to `coxph`.

Value

A list of concordance and the model fit with the training data.

---

**nhanes_example**

*Example Data from National Health and Nutrition Examination Survey ('NHANES')*

Description

Example Data from National Health and Nutrition Examination Survey ('NHANES')

Usage

```
nhanes_example
```

Format

A data frame with 7 columns, which are:

- **SEQN**: ID of participant
- **mortstat**: Mortality status, 1-died, 0-censored
- **event_time_years**: Time observed
- **WTMEC4YR_norm**: Weights normalized for survey
- **gender**: Gender
- **age_years_interview**: Age in years at interview
- **education_adult**: Educational status
Index

* datasets
  nhanes_example, 5

cforward, 2

cforward_one (cforward), 2

concordancefit, 3, 5

coxph, 3, 5

estimate_concordance, 4

make_folds (cforward), 2

nhanes_example, 5

Surv, 2, 5