Package ‘ahMLE’

January 12, 2022

Type Package
Title Methods for the Additive Hazard Model
Version 1.20
Date 2021-12-22
Author Chengyuan Lu
Maintainer Chengyuan Lu <c.lu@lumc.nl>
Description Methods for fitting additive hazards model.
       Perform the maximum likelihood method as well as the traditional Aalen's method for estimating the additive hazards model.
License GPL (>= 2)
Depends R (>= 3.1.0), survival
Imports Rcpp (>= 1.0.1), Matrix, invGauss
LinkingTo Rcpp, RcppArmadillo, Matrix
SystemRequirements C++11
Encoding UTF-8
Suggests knitr, rmarkdown
VignetteBuilder knitr
RoxygenNote 7.1.0
NeedsCompilation yes
Repository CRAN
Date/Publication 2022-01-12 21:52:44 UTC

R topics documented:

ah ................................................................. 2
Calc_Cbeta ..................................................... 3

Index 4
**ah**

### Description

This function offers the methods to fit the additive hazards model, including Aalen’s method and Maximum likelihood method.

### Usage

```r
ah(
    formula = formula(data),
    data = sys.parent(),
    matrix_domain = NULL,
    progbar = FALSE,
    method = "ml_opt",
    scale = TRUE,
    startedge = NULL
)
```

### Arguments

- **formula**: A formula, the dependent variable must be of type Surv in the survival package.
- **data**: A data frame with the covariates mentioned in the formula stored.
- **matrix_domain**: A matrix describing the domain to find the maximum likelihood. The default constraint matrix guarantees the hazards to be positive for all possible covariates.
- **progbar**: A logical value, shows the progress bar if it is TRUE, hide the progress bar if FALSE, default value is FALSE.
- **method**: A string with values "aalen", "ml_opt", "ml_enum", "ml_asc" and "ml_desc". Default value is "ml_opt". "aalen" represents the Aalen’s method. "ml_opt" is the default method with respect to the default constraint matrix. "ml_enum","ml_asc" and "ml_desc" represents to the naive method, ascending method and descending method. Please check the referee for details.
- **scale**: A logical value, scales the input data in the interval [0,1] if it is TRUE. Default value is TRUE.
- **startedge**: a vector which satisfies the domain condition. Only used for the ascending method.

### Value

A data frame, containing the coefficients (beta) at each time point and the cumulative beta at each time point.

### References

Calc_Cbeta

Examples
X1 = rnorm(100); X2 = rnorm(100)
Survival_Time = rep(0,100)
U = runif(100, min =0, max =1)
for (i in 1:100){Survival_Time[i] = sqrt((-2*log(U[i])/(0.3*X1[i] + 0.7*X2[i])))}
tcens = runif(100, 2.5, 7.5)
time = pmin(Survival_Time, tcens)
event = as.numeric(Survival_Time<=tcens)
Data = data.frame(time = time, X1 = X1, X2 = X2, event = event)
Result = ah(Surv(time = time, event = event)~ X1 + X2, Data)

Description
Computing cumulative beta from beta (the magnitude of the jumps of the coefficient function)

Usage
Calc_Cbeta(beta)

Arguments
beta The dataframe storing the magnitude of the jumps beta of the coefficients of the additive hazard. The first column should be the survival time.

Value
Return the cumulative beta by the giving beta.

Examples
beta = data.frame(time = c(1,2,3,4), beta = c(5,6,7,8))
cbeta = Calc_Cbeta(beta)
Index

ah, 2
Calc_Cbeta, 3