Package ‘plac’

August 29, 2016

Type Package
Title A Pairwise Likelihood Augmented Cox Estimator for Left-Truncated Data
Version 0.1.1
Date 2016-04-29
Description A semi-parametric estimation method for the Cox model with left-truncated data using augmented information from the marginal of truncation times.
Depends R (>= 3.2.0), survival (>= 2.38-3)
License GPL-3
LazyData TRUE
Imports Rcpp (>= 0.12.1),
URL https://github.com/942kid/plac
BugReports https://github.com/942kid/plac/issues
LinkingTo Rcpp, RcppEigen
RoxygenNote 5.0.1
Suggests testthat
NeedsCompilation yes
Author Fan Wu [aut, cre]
Maintainer Fan Wu <fannwu@umich.edu>
Repository CRAN
Date/Publication 2016-04-30 00:15:59

R topics documented:

plac-package ................................................................. 2
cum.haz ................................................................. 3
PLAC ................................................................. 3
PLAC_TD ............................................................... 5
plac-package

A Package for Computing the Pairwise Likelihood Augmented Cox Estimator for Left-Truncated Data.

Description

This package provides both lower-level C++ functions (PLAC_TI(), PLAC_TV() and PLAC_TvR()) and an R wrapper function PLAC() to calculate the pairwise likelihood augmented Cox estimator for left-truncated survival data as proposed by Wu et al. (2015).

Wrapper Function PLAC()

This R wrapper function calls different C++ function depending on the covariate types data has.

C++ Functions

The three C++ functions PLAC_TI(), PLAC_TV() and PLAC_TvR() provide a direct interface to the algorithm in case that users need to supply more flexible time-dependent coavriates other than indicator functions.

References

Calculate the Values of the cumulative Hazard at Fixed Times

Description

Calculate the Values of the cumulative Hazard at Fixed Times

Usage

cum.haz(est, t.eval = c(0.25, 0.75))

Arguments

est an object of the class plac.fit.
t.eval time points at which the Lambda(t) is evaluated (for both conditional approach and the PLAC estimator).

Value

a list containing the estimates and SEs of Lambda(t) for both conditional approach and the PLAC estimator.

Examples

dat1 = sim.ltrc(n = 100)$dat
est = PLAC(ltrc.formula = Surv(As, Ys, Ds) ~ Z1 + Z2,
    ltrc.data = dat1, td.type = "none")
H = cum.haz(est, t.eval = seq(0.1, 0.9, 0.1))
H$V
H$se

PLAC Calculate the PLAC estimator when a time-dependent indicator presents

Description

Both a conditional approach Cox model and a pairwise likelihood augmented estimator are fitted and the corresponding results are returned in a list.

Usage

PLAC(ltrc.formula, ltrc.data, id.var = "ID", td.var = NULL,
    td.type = "none", t.jump = NULL, init.val = NULL, max.iter = 100,
    print.result = TRUE, ...)

Arguments

ltrc.formula a formula of of the form Surv(A, Y, D) ~ Z, where Z only include the time-invariate covariates.
ltrc.data a data.frame of the LTRC dataset including the responses, time-invariate covariates and the jump times for the time-depenent covariate.
id.var a name of the subject id in data.
td.var a name of the time-dependent covariate in the output.
td.type the type of the time-dependent covariate. Either one of c("none", "independent", "post-trunc", "pre-post-trunc"). See Details.
t.jump a name of the jump time variable in data.
init.val a list of the initial values of the coefficients and the baseline hazard function for the PLAC estimator.
max.iter the maximal number of iteration for the PLAC estimator
print.result logical, if a brief summary of the regression coefficient estimates should be printed out.
... other arguments

Details

ltrc.formula should have the same form as used in coxph(); e.g., Surv(A, Y, D) ~ Z1 + Z2. where A is the truncation time (tstart), Y is the survival time (tstop) and D is the status indicator (event). td.type is used to determine which C++ function will be invoked: either PLAC_TI (if td.type = "none"), PLAC_TD (if td.type = "independent") or PLAC_TDR (if td.type %in% c("post-trunc", "pre-post-trunc")). For td.type = "post-trunc", the pre-truncation values for the time-dependent covariate will be set to be zero for all subjects.

Value

a list of model fitting results for both conditional approach and the PLAC estimators.

Event.Time Ordered distinct observed event times
b Regression coefficients estimates
se.b Model-based SEs of the regression coefficients estimates
h0 Estimated cumulative baseline hazard function
se.h0 Model-based SEs of the estimated cumulative baseline hazard function
sandwich The sandwich estimator for (beta, lambda)
k The number of iteration for used for the PLAC estimator
summ A brief summary of the covariates effects

References


Examples

# When only time-invariant covariates are involved
dat1 = sim.ltrc(n = 50)$dat
PLAC(ltrc.formula = Surv(As, Ys, Ds) ~ Z1 + Z2, 
   ltrc.data = dat1, td.type = "none")

# When there is a time-dependent covariate that is independent of the truncation time
dat2 = sim.ltrc(n = 50, time.dep = TRUE, 
   distr.A = "binomial", p.A = 0.8, Cmax = 5)$dat
PLAC(ltrc.formula = Surv(As, Ys, Ds) ~ Z, 
   ltrc.data = dat2, td.type = "independent", 
   td.var = "Zv", t.jump = "zeta")

# When there is a time-dependent covariate that depends on the truncation time
dat3 = sim.ltrc(n = 50, time.dep = TRUE, Zv.depA = TRUE, Cmax = 5)$dat
PLAC(ltrc.formula = Surv(As, Ys, Ds) ~ Z, 
   ltrc.data = dat3, td.type = "post-trunc", 
   td.var = "Zv", t.jump = "zeta")

---

**PLAC_TD**

C++ Function for Solving the PLAC Estimator. (with time-dependent covariates independent of A^*)

Description

C++ Function for Solving the PLAC Estimator. (with time-dependent covariates independent of A^*)

Usage

`PLAC_TD(Z, ZFV_, X, W, Ind1, Ind2, Dn, b, h, K = 100L)`

Arguments

- **Z**: matrix for all the covariates history.
- **ZFV_**: matrix for all covariates at the each individual’s observed survival time.
- **X**: the response matrix (As, Xs, Ds).
- **W**: the ordered observed event times.
- **Ind1**: risk-set indicators.
- **Ind2**: truncation pair indicators.
- **Dn**: number of ties at each observed event time.
- **b**: initial values of the regression coefficients.
- **h**: initial values of the baseline hazard function.
- **K**: maximal iteration number, the default is K = 100.

Value

list of model fitting results for both conditional approach and the PLAC estimator.
### Description

C++ Function for Solving the PLAC Estimator. (with time-dependent covariates depending on $A^{**}$)

### Usage

```c
PLAC_TDR(ZF, ZFV_, Z, W, Ind1, Ind2, Dn, b, h, K = 100L)
```

### Arguments

- **ZF**: matrix for all the time-invariant covariates.
- **ZFV_**: matrix for all covariates at the each individual’s observed survival time.
- **Z**: matrix for all the covariates history.
- **X**: the response matrix ($A_s, X_s, D_s$).
- **W**: the ordered observed event times.
- **Ind1**: risk-set indicators.
- **Ind2**: truncation pair indicators.
- **Dn**: number of ties at each observed event time.
- **b**: initial values of the regression coefficients.
- **h**: initial values of the baseline hazard function.
- **K**: maximal iteration number, the default is $K = 100$.

### Value

list of model fitting results for both conditional approach and the PLAC estimator.

---

### Description

C++ Function for Solving the PLAC Estimator. (with time-invariant covariates only)

### Usage

```c
PLAC_TI(Z, X, W, Ind1, Ind2, Dn, b, h, K = 100L)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>matrix for all the covariates history.</td>
</tr>
<tr>
<td>X</td>
<td>the response matrix (As, Xs, Ds).</td>
</tr>
<tr>
<td>W</td>
<td>the ordered observed event times.</td>
</tr>
<tr>
<td>Ind1</td>
<td>risk-set indicators.</td>
</tr>
<tr>
<td>Ind2</td>
<td>truncation pair indicators.</td>
</tr>
<tr>
<td>Dn</td>
<td>number of ties at each observed event time.</td>
</tr>
<tr>
<td>b</td>
<td>initial values of the regression coefficients.</td>
</tr>
<tr>
<td>h</td>
<td>initial values of the baseline hazard function.</td>
</tr>
<tr>
<td>K</td>
<td>maximal iteration number, the default is $K = 100$.</td>
</tr>
</tbody>
</table>

Value

list of model fitting results for both conditional approach and the PLAC estimator.

---

**plr**

Perform the paired log-rank test.

---

Description

Perform the paired log-rank test on the truncation times and the residual survival times to check the stationarity assumption (uniform truncation assumption) of the left-truncated right-censored data.

Usage

```r
plr(dat, A.name = "As", Y.name = "Ys", D.name = "Ds")
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>a data.frame of left-truncated right-censored data.</td>
</tr>
<tr>
<td>A.name</td>
<td>the name of the truncation time variable in dat.</td>
</tr>
<tr>
<td>Y.name</td>
<td>the name of the survival time variable in dat.</td>
</tr>
<tr>
<td>D.name</td>
<td>the name of the event indicator in dat.</td>
</tr>
</tbody>
</table>

Value

a list containing the test statistic and the p-value of the paired log-rank test.

References


Examples

```r
dat = sim.ltrc(n = 100, distr.A = "weibull")$dat
plr(dat)
```
### PwInd

Generate truncation-pair indicators

**Description**
Generate truncation-pair indicators

**Usage**
PwInd(X, W)

**Arguments**
- \( X \) the response matrix (As, Xs, Ds).
- \( W \) the ordered observed event times.

**Value**
the truncation-pair indicators of the form \( I(w_k \leq A_i) - I(w_k \leq XA_j) \).

### SgInd

Generate risk-set indicators

**Description**
Generate risk-set indicators

**Usage**
SgInd(X, W)

**Arguments**
- \( X \) the response matrix (As, Xs, Ds).
- \( W \) the ordered observed event times.

**Value**
risk-set indicators \( Y_i(w_k) \) of the form \( I(A_i \leq w_k \leq X_i) \).
Generate left-truncated (and right-censored) data from the Cox model.

Description

Various baseline survival functions and truncation distribution are available. Censoring rate can be designated through tuning the parameter Cmax; Cmax = Inf means no censoring.

Usage

sim.ltrc(n = 200, b = c(1, 1), time.dep = FALSE, Zv.depA = FALSE, distr.T = "weibull", shape.T = 2, scale.T = 1, meanlog.T = 0, sdlog.T = 1, distr.A = "weibull", shape.A = 1, scale.A = 5, p.A = 0.3, Cmax = Inf, fix.seed = NULL)

Arguments

n the sample size.
b a numeric vector for true regression coefficients.
time.dep logical, whether there is the time-dependent covariate (only one indicator function Zv = I(t >= zeta) is supported); the default is FALSE.
Zv.depA logical, whether the time-dependent covariate Zv depends on A^* (the only form supported is Zv = I(t >= zeta + A^*)); the default is FALSE.
distr.T the baseline survival time (T*) distribution ("exp" or "weibull").
shape.T the shape parameter for the Weibull distribution of T*.
scale.T the scale parameter for the Weibull distribution of T*.
meanlog.T the mean for the log-normal distribution of T*.
sdlog.T the sd for the log-normal distribution of T*.
distr.A the baseline truncation time (A^*) distribution: either of "weibull" (the default), "unif" (Length-Biased Sampling), "binomial" or "dunif"). Note: If distribution name other than these are provided, "unif" will be used.
shape.A the shape parameter for the Weibull distribution of A^*.
scale.A the scale parameter for the Weibull distribution of A^*.
p.A the success probability for the binomial distribution of A^*.
Cmax the upper bound of the uniform distribution of the censoring time (C).
fix.seed an optional random seed for simulation.

Value

a list with a data.frame containing the observed survival times (Ys), the observed truncation times (As), the event indicator (Ds) and the covariates (Zs); a vector of certain quantiles of Ys (taus); the censoring proportion (PC) and the truncation proportiona (PT).
Examples

# With time-invariant covariates only
sim1 = sim.ltrc(n = 100)
head(sim1$dat)

# With one time-dependent covariate
sim2 = sim.ltrc(n = 100, time.dep = TRUE,
    distr.A = "binomial", p.A = 0.8, Cmax = 5)
head(sim2$dat)

# With one time-dependent covariate with dependence on the truncation time
sim3 = sim.ltrc(n = 100, time.dep = TRUE, zv.depA = TRUE, Cmax = 5)
head(sim3$dat)

TvInd

Generate time-dependent covariate indicators

Description
Generate time-dependent covariate indicators

Usage
TvInd(zeta, W)

Arguments

zeta  
the change point (jump time) of Z_v(t).

W  
the ordered observed event times.

Value

the time-dependent covariate of the form Z_v(t) = I(w_k > zeta).

TvInd2

Generate time-dependent covariate indicators

Description
Generate time-dependent covariate indicators

Usage
TvInd2(eta, zeta, W)
Arguments

- \(\eta\) a random variable of the \(Z_v(t)\) value before the change point.
- \(\zeta\) the change point (jump time).
- \(\mathcal{W}\) the ordered observed event times.

Value

the time-dependent covariate indicators of the form \(Z_v(t) = \eta \cdot I(\mathcal{W}_k \leq \zeta)\).
Index

cum.haz, 3

PLAC, 3
plac-package, 2
PLAC_TD, 5
PLAC_TDR, 6
PLAC_TII, 6
plr, 7
PwInd, 8

SgInd, 8
sim.ltrc, 9

TvInd, 10
TvInd2, 10