

Package ‘IntCens’

February 23, 2025

Type Package

Title Non- and Semi-Parametric Analysis of Interval-Censored Data

Version 0.2.0

Author Lu Mao

Maintainer Lu Mao <lmao@biostat.wisc.edu>

Description Implements non- and semi-parametric methods for analyzing interval-censored data. This includes iterative estimation routines for proportional hazards and proportional odds models, as well as nonparametric maximum likelihood estimation.

License CC BY 4.0

Encoding UTF-8

LazyData true

URL <https://github.com/lmaowisc/IntCens>, <https://lmaowisc.github.io/IntCens/>

Depends R (>= 3.5.0)

RoxygenNote 7.3.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Contents

bcos	2
CM	2
CSD.f2	3
icsurvfit	4
order_rank	6
plot.icsurvfit	7
print.icsurvfit	7
Index	9

bcos

Breast Cosmesis Data

Description

A dataset containing interval-censored observations and a treatment indicator (radiation vs. radiation plus chemotherapy). The variables are as follows:

Usage

```
data(bcos)
```

Format

A data frame with 94 rows and 3 variables:

left Left endpoints of the censoring intervals (numeric).

right Right endpoints of the censoring intervals (numeric).

treatment A factor with levels "Rad" (radiation only) or "RadChem" (radiation + chemotherapy).

Details

These data are often used to illustrate methods for interval-censored survival analysis. Observations with Inf in the right endpoint represent right-censored data.

Examples

```
data(bcos)
head(bcos)
```

CM

Compute Convex Minorant

Description

Computes a piecewise-linear convex minorant of two numeric vectors G and Q by iteratively identifying minimum slopes and assigning them to the output vector y. This method is sometimes used in isotonic or shape-constrained regression contexts.

Usage

```
CM(G, Q)
```

Arguments

G Numeric vector of x -coordinates (must be the same length as Q).

Q Numeric vector of y -coordinates (must be the same length as G).

Details

Internally, the function calculates slopes for segments between points in G and picks the minimum slope iteratively, ensuring the resulting piecewise-linear function is convex.

Value

A numeric vector y of length length(G), giving the slopes or values associated with the convex minorant.

CSD.f2	<i>Compute Score Components for Interval-Censored Data</i>
--------	--

Description

Computes partial derivatives or score components for interval-censored data, given current estimates of baseline functions and a regression parameter vector beta. This function is typically part of an iterative algorithm for semiparametric estimation.

Usage

CSD.f2(Hf, beta, y, index, delta, gamma, Z)

Arguments

Hf	A list containing baseline functions or related objects (e.g., H, H1, H2).
beta	A numeric vector of regression coefficients for the covariates in Z.
y	A numeric vector (often the current estimate of the cumulative baseline hazard or a similar function) of length p.
index	An n x 2 matrix of indices referencing positions in y for each subject.
delta	Numeric (0/1) event indicator for the left endpoint U.
gamma	Numeric (0/1) event indicator for the right endpoint V.
Z	A matrix or data frame of covariates, with n rows (one per subject).

Details

This function uses auxiliary routines (e.g., FdG.f(), eta1G.f(), eta2G.f(), zeta1.f(), etc.) to compute partial derivatives of the log-likelihood or estimating equations with respect to y.

Value

- A list with the following components:
- G A numeric vector (length p) of cumulative increments related to gamma.
 - Q A numeric vector (length p) of combined increments (e.g., dW + y[i] * dG).

Examples

```
# (Not run) Example with placeholder data:
# Hf <- list(H = NULL, H1 = NULL, H2 = NULL)
# beta <- c(0.1, -0.2)
# y <- rep(0.05, 5)
# index <- matrix(c(1, 2, 2, 3), nrow = 2, byrow = TRUE)
# delta <- c(1, 0)
# gamma <- c(0, 1)
# Z <- matrix(c(1, 0, 0, 1), nrow = 2)
# CSD.f2(Hf, beta, y, index, delta, gamma, Z)
```

 icsurvfit

Fit Interval-Censored Model

Description

Fits an interval-censored model of various types (NP, PO, or PH) by iterating over baseline and regression parameter updates.

Usage

```
icsurvfit(L, R, Z, model = "NP", maxiter = 500)
```

Arguments

L	Left endpoints (may contain NA, -Inf).
R	Right endpoints (may contain NA, Inf).
Z	A matrix or data frame of covariates.
model	One of "NP", "PO", or "PH". Defaults to "NP" for nonparametric (no covariates).
maxiter	Maximum number of iterations for the fitting algorithm.

Details

NP Nonparametric (no covariate effects).

PO Proportional odds.

PH Proportional hazards.

Internally, this function sets up the data, identifies which observations are left-, right-, or interval-censored, and then calls various helper routines (e.g., [profile1](#), [NR2](#)) to perform the iterative fitting.

Value

A list (class "icsurvfit") with components:

beta	Estimated regression coefficients (if model="PO" or "PH").
y	Baseline function estimates at each time index.
t	Ordered time points at which y is evaluated.
var	Estimated variance-covariance matrix of beta (if applicable).
j	Number of iterations used.

model The specified model type.

conv Logical indicating whether the algorithm converged.

varnames Covariate names.

call Matched call.

Examples

```
# Simple example
data(bcos)

n <- nrow(bcos) # sample size
L <- bcos$left + rnorm(n, 0, 0.0001) # add random noise
R <- bcos$right + rnorm(n, 0, 0.0001)
Z <- as.numeric(bcos$treatment == "RadChem")

# Nonparametric
obj_np <- icsurvfit(L, R)

# Cox model (PH)
obj_ph <- icsurvfit(L, R, Z, model = "PH")
obj_ph
#> Call:
#> icsurvfit(L = L, R = R, Z = Z, model = "PH")
#>
#> NPMLE of proportional hazards for interval-censored data:
#>
#> ICM algorithm converges in 20 iterations.
#>
#> Maximum Likelihood Estimates for Regression parameters:
#>
#> Estimate StdErr z.value p.value
#> [1,] 0.78883 0.29344 2.6882 0.007183 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Proportional odds model (PO)
obj_po <- icsurvfit(L, R, Z, model = "PO")
obj_po
#> Call:
#> icsurvfit(L = L, R = R, Z = Z, model = "PO")
#>
#> NPMLE of proportional odds for interval-censored data:
#>
#> ICM algorithm converges in 20 iterations.
#>
#> Maximum Likelihood Estimates for Regression parameters:
#>
#> Estimate StdErr z.value p.value
#> [1,] 0.89529 0.41137 2.1764 0.02953 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

order_rank

Order and Rank Interval-Censored Data

Description

Processes interval-censored data (and corresponding indicators) to produce an ordered set of exam times and flags indicating censoring status. It also adjusts certain delta and gamma values based on constraints so that the first observation is always an event time, and the last observation is always right-censored.

Usage

```
order_rank(delta, gamma, U, V, Z)
```

Arguments

delta	Numeric (0/1). Event indicator for left endpoint U.
gamma	Numeric (0/1). Event indicator for right endpoint V.
U	Numeric vector of left endpoints of censoring intervals (may contain NA).
V	Numeric vector of right endpoints of censoring intervals (may contain NA).
Z	A matrix (or data frame) of covariates, with one row per subject.

Details

This function ensures that the earliest recorded time in `t` corresponds to a left endpoint that is truly an event, while the latest recorded time is right-censored. It also removes observations that are no longer valid after these adjustments. The vector `flag` is used internally to mark observation types (1 = event, 2 = left-censored, 3 = both, 4 = right-censored).

Value

A list with the following components:

`t` A condensed, ordered set of exam times.

`index` A matrix of dimension $n \times 2$, giving the ranks (indices) for U and V after ordering.

`data` A matrix (or data frame) containing the updated delta, gamma, U, V, and Z values.

Examples

```
# (Not run) Example usage (requires suitable data)
# delta <- c(1, 0, 0)
# gamma <- c(0, 1, 0)
# U <- c(2, 2, 3)
# V <- c(NA, 4, 5)
# Z <- matrix(rnorm(9), nrow = 3)
# order_rank(delta, gamma, U, V, Z)
```

plot.icsurvfit

*Plot Method for icsurvfit Objects***Description**

Plots the estimated survival function from an `icsurvfit` object. If `model="NP"`, the nonparametric estimate is plotted. For `model="PO"` or `"PH"`, the survival curve for a specific covariate value `z` can be overlaid. By default, it plots a step function.

Usage

```
## S3 method for class 'icsurvfit'
plot(
  object,
  z = NULL,
  xlab = "Time",
  ylab = "Survival rate",
  lty = 1,
  frame.plot = FALSE,
  add = FALSE,
  ylim = c(0, 1),
  ...
)
```

Arguments

<code>object</code>	An object of class <code>"icsurvfit"</code> returned by icsurvfit .
<code>z</code>	Numeric (vector of) covariate values for which to plot the survival curve. If <code>NULL</code> (default) and <code>model</code> is nonparametric, plots the baseline curve.
<code>xlab, ylab</code>	Axis labels.
<code>lty</code>	Line type. Default is 1.
<code>frame.plot</code>	Logical indicating whether to draw a box around the plot.
<code>add</code>	Logical indicating whether to add the curve to an existing plot. If <code>FALSE</code> (default), a new plot is created.
<code>ylim</code>	Numeric vector of length 2 specifying the <i>y</i> -axis range.
<code>...</code>	Additional graphical parameters passed to plot or lines .

print.icsurvfit

*Print Method for icsurvfit Objects***Description**

Prints a summary of the fitted model, including the call, convergence status, estimated survival function (for `model="NP"`), or estimated regression coefficients (for `model="PO"` or `"PH"`).

Usage

```
## S3 method for class 'icsurvfit'  
print(object, ...)
```

Arguments

<code>object</code>	An object of class "icsurvfit" returned by icsurvfit .
<code>...</code>	Further arguments passed to or from other methods. Not currently used.

Index

* datasets

bcos, [2](#)

bcos, [2](#)

CM, [2](#)

CSD.f2, [3](#)

icsurvfit, [4](#), [7](#), [8](#)

lines, [7](#)

NR2, [4](#)

order_rank, [6](#)

plot, [7](#)

plot.icsurvfit, [7](#)

print.icsurvfit, [7](#)

profile1, [4](#)