Package ‘SmoothHazard’

July 12, 2017

Title Estimation of Smooth Hazard Models for Interval-Censored Data with Applications to Survival and Illness-Death Models

Version 1.4.1

Author Celia Touraine, Pierre Joly, Thomas A. Gerds

Description Estimation of two-state (survival) models and irreversible illness-death models with possibly interval-censored, left-truncated and right-censored data. Proportional intensities regression models can be specified to allow for covariates effects separately for each transition. We use either a parametric approach with Weibull baseline intensities or a semi-parametric approach with M-splines approximation of baseline intensities in order to obtain smooth estimates of the hazard functions. Parameter estimates are obtained by maximum likelihood in the parametric approach and by penalized maximum likelihood in the semi-parametric approach.

Depends R (>= 1.9.1), prodlim (>= 1.4.9)

Imports lava (>= 1.4.1), mvtnorm (>= 1.0-3)

Maintainer Thomas Alexander Gerds <tag@biostat.ku.dk>

License GPL (>= 2)

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-07-12 10:30:39 UTC

R topics documented:

idm ................................................................. 2
idmModel .......................................................... 7
intensity .......................................................... 9
Paq1000 .......................................................... 11
plot.idm .......................................................... 12
plot.shr ......................................................... 13
predict.idm ....................................................... 15
print.idm ......................................................... 17
Fit an illness-death model

**Description**

Fit an illness-death model using either a semi-parametric approach (penalized likelihood with an approximation of the transition intensity functions by linear combination of M-splines) or a parametric approach (specifying Weibull distributions on the transition intensities). Left-truncated, right-censored, and interval-censored data are allowed. State 0 corresponds to the initial state, state 1 to the transient one, state 2 to the absorbant one. The allowed transitions are: 0 \to 1, 0 \to 2 and 1 \to 2.

**Usage**

```r
idm(formula01, formula02, formula12, data, maxiter = 200, eps = c(5, 5, 3),
    n.knots = c(7, 7, 7), knots = "equidistant", CV = FALSE,
    kappa = c(1000000, 500000, 200000), method = "Weib", conf.int = 0.95,
    print.iter = FALSE, subset = NULL, na.action = na.fail)
```

**Arguments**

- **formula01**: A formula specifying a regression model for the 0 \to 1 transition from the initial state to the transient state of the illness-death model. The right hand side of the formula specifies the covariate terms, and the left hand side must be an event history object as returned by the function `Hist`.
- **formula02**: A formula specifying a regression model for the 0 \to 2 transition from the initial state to the absorbing state. The left hand side must be equal to the left hand side of `formula01`. If missing it is set to `formula01`.
- **formula12**: A formula specifying a regression model for the 1 \to 2 transition from the transient state to the absorbing state. The operator is not required. If missing it is set to `formula01`.
- **data**: A data frame in which to interpret the variables of `formula01`, `formula02` and `formula12`.
- **maxiter**: Maximum number of iterations. The default is 200.
eps

A vector of 3 integers >0 used to define the power of three convergence criteria: 1. for the regression parameters, 2. for the likelihood, 3. for the second derivatives. The default is c(5,5,3) which is translated into convergence if the respective values change less then $10^{-5}$ (for regression parameters and likelihood) and $10^{-3}$ for the second derivatives between two iterations.

n.knots

For method="Splines" only, a vector of length 3 specifying the number of knots, one for each transition, for the M-splines estimate of the baseline intensities in the order $0 \rightarrow 1, 0 \rightarrow 2, 1 \rightarrow 2$. The default is c(7,7,7). When knots are specified as a list this argument is ignored. The algorithm needs at least 5 knots and at most 20 knots.

knots

Argument only active for the penalized likelihood approach method="Splines". There are three ways to control the placement of the knots between the smallest and the largest of all time points:

- knots="equidistant" Knots are placed with same distance on the time scale.
- knots="quantiles" Knots are placed such that the number of observations is roughly the same between knots.
- knots=list() List of 1 or 2 or three vectors. The list elements are the actual placements (timepoints) of the knots for the M-spline. The list may contain one vector of placements for each transition in the order $0 \rightarrow 1, 0 \rightarrow 2, 1 \rightarrow 2$. If only vector is specified the knots are used for all transitions. If only 2 vectors are specified, the knots for the $0 \rightarrow 1$ transition are also used for the $1 \rightarrow 2$ transition.

The algorithm needs at least 5 knots and allows no more than 20 knots.

CV

Binary variable equals to 1 when search (by approximated cross validation) of the smoothing parameters kappa and 0 otherwise. Argument for the penalized likelihood approach. The default is 0.

kappa

Argument only active for the penalized likelihood approach method="Splines". A vector with 3 positive values (smoothing parameters), one for each transition, in the order $0 \rightarrow 1, 0 \rightarrow 2$ and $1 \rightarrow 2$. If CV=1 these are used as starting values for a cross validation search to optimize kappa.

method

type of estimation method: "Splines" for a penalized likelihood approach with approximation of the transition intensities by M-splines, "Weib" for a parametric approach with a Weibull distribution on the transition intensities. Default is "Weib".

conf.int

Level of confidence pointwise confidence intervals of the transition intensities, i.e., a value between 0 and 1, the default is 0.95. The default is also used when conf.int=TRUE. To avoid computation of confidence intervals, set conf.int to FALSE or NULL.

print.iter

boolean parameter. Equals to TRUE to print the likelihood during the iteration process, FALSE otherwise. Default is FALSE. This option is not running on Windows.

subset

expression indicating the subset of the rows of data to be used in the fit. All observations are included by default.
na.action how NAs are treated. The default is first, any na.action attribute of data, second
a na.action setting of options, and third 'na.fail' if that is unset. The 'factory-
fresh' default is na.omit. Another possible value is NULL.

Details

The estimated parameters are obtained using the robust Marquardt algorithm (Marquardt, 1963)
which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm.

Value

call the call that produced the result.
coef regression parameters.
loglik vector containing the log-likelihood without and with covariate.
cv vector containing the convergence criteria.
niter number of iterations.
converged integer equal to 1 when the model converged, 2, 3 or 4 otherwise.
modelPar Weibull parameters.
N number of subjects.
events1 number of events 0 –> 1.
events2 number of events 0 –> 2 or 0 –> 1 –> 2.
NC vector containing the number of covariates on transitions 0 –> 1, 0 –> 2, 1 –> 2.
responseTrans model response for the 0 –> 1 transition. Hist or Surv object.
responseAbs model response for the 0 –> 2 transition. Hist or Surv object.
time times for which transition intensities have been evaluated for plotting. Vector
in the Weibull approach. Matrix in the penalized likelihood approach for which
the columns corresponds to the transitions 0 –> 1, 1 –> 2, 0 –> 2.
intensity01 matched values of the intensities for transition 0 –> 1.
lowerIntensity01 lower confidence intervals for the values of the intensities for transition 0 –> 1.
upperIntensity01 upper confidence intervals for the values of the intensities for transition 0 –> 1.
intensity02 matched values of the intensities for transition 0 –> 2.
lowerIntensity02 lower confidence intervals for the values of the intensities for transition 0 –> 2.
upperIntensity02 upper confidence intervals for the values of the intensities for transition 0 –> 2.
intensity12 matched values of the intensities for transition 1 –> 2.
lowerIntensity12 lower confidence intervals for the values of the intensities for transition 1 –> 2.
upperIntensity12 upper confidence intervals for the values of the intensities for transition 1 –> 2.
RR vector of relative risks.
V variance-covariance matrix derived from the Hessian of the log-likelihood if using method="Weib" or, from the Hessian of the penalized log-likelihood if using method="Splines".

se standart errors of the regression parameters.

Xnames01 names of covariates on 0 -> 1.

Xnames02 names of covariates on 0 -> 2.

Xnames12 names of covariates on 1 -> 2.

knots01 knots to approximate by M-splines the intensity of the 0 -> 1 transition.

knots02 knots to approximate by M-splines the intensity of the 0 -> 2 transition.

knots12 knots to approximate by M-splines the intensity of the 1 -> 2 transition.

nknots01 number of knots on transition 0 -> 1.

nknots02 number of knots on transition 0 -> 2.

nknots12 number of knots on transition 1 -> 2.

theta01 square root of splines coefficients for transition 0 -> 1.

theta02 square root of splines coefficients for transition 0 -> 2.

theta12 square root of splines coefficients for transition 1 -> 2.

CV a binary variable equals to 1 when search of the smoothing parameters kappa by approximated cross-validation, 1 otherwise. The default is 0.

kappa vector containing the smoothing parameters for transition 0 -> 1, 0 -> 2, 1 -> 2 used to estimate the model by the penalized likelihood approach.

cVCrit cross validation criteria.

DoF degrees of freedom of the model.

na.action observations deleted if missing values.

Author(s)

R: Celia Touraine <CeliaTouraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <PierreJoly@isped.u-bordeaux2.fr>

References


See Also

print.idm summary.idm predict.idm
Examples

```r
library(lava)
library(prodlim)
set.seed(17)
d <- simulateIDM(100)
# right censored data
fitRC <- idm(formula0=Hist(time=observed.illtime,event=seen.ill)-X1+X2,
 formula1=Hist(time=observed.lifetime,event=seen.exit)-X1+X2,
 formula2=Hist(time=observed.lifetime,event=seen.exit)-X1+X2, data=d, conf.int=FALSE)
fitRC

## Not run:
set.seed(17)
d <- simulateIDM(300)
fitRC.splines <- idm(formula0=Hist(time=observed.illtime,event=seen.ill)-X1+X2,
 formula1=Hist(time=observed.lifetime,event=seen.exit)-X1+X2,
 formula2=Hist(time=observed.lifetime,event=seen.exit)-X1+X2, data=d, conf.int=FALSE,method="splines")

## End(Not run)
# interval censored data
fitIC <- idm(formula0=Hist(time=list(L,R),event=seen.ill)-X1+X2,
 formula1=Hist(time=observed.lifetime,event=seen.exit)-X1+X2,
 formula2=Hist(time=observed.lifetime,event=seen.exit)-X1+X2, data=d, conf.int=FALSE)
fitIC

## Not run:

data(Paq1000)

# Illness-death model with certif on the 3 transitions
# Weibull parametrization and likelihood maximization
fit.weib <- idm(formula0=Hist(time=t,event=death,entry=e)-certif,
 formula1=Hist(time=list(l,r),event=dementia)-certif,
 data=Paq1000)

# Illness-death model with certif on transitions 01 and 02
# Splines parametrization and penalized likelihood maximization
fit.splines <- idm(formula0=Hist(time=t,event=death,entry=e)-certif,
 formula1=Hist(time=list(l,r),event=dementia)-certif,
 formula2=-1,
 method="Splines",
 data=Paq1000)
fit.weib
summary(fit.splines)

## End(Not run)
```
**idmModel**

*Generate illness-death model objects*

**Description**

Function to generate an illness-death model for simulation.

**Usage**

```r
idmModel(scale.illtime = 1/100, shape.illtime = 1, scale.lifetime = 1/100,
shape.lifetime = 1, scale.waittime = 1/100, shape.waittime = 1,
scale.censtime = 1/100, shape.censtime = 1, n.inspections = 5,
schedule = 10, punctuality = 5)
```

**Arguments**

- `scale.illtime`: Weibull scale for latent illness time
- `shape.illtime`: Weibull shape for latent illness time
- `scale.lifetime`: Weibull scale for latent life time
- `shape.lifetime`: Weibull shape for latent life time
- `scale.waittime`: Weibull scale for latent life time
- `shape.waittime`: Weibull shape for latent life time
- `scale.censtime`: Weibull scale for censoring time
- `shape.censtime`: Weibull shape for censoring time
- `n.inspections`: Number of inspection times
- `schedule`: Mean of the waiting time between adjacent inspections.
- `punctuality`: Standard deviation of waiting time between inspections.

**Details**

Based on the functionality of the lava PACKAGE the function generates a latent variable model (latent illtime, waittime and lifetime) and censoring mechanism (censtime, inspection1,inspection2,...,inspectionK).

The function `sim.idmModel` then simulates right censored lifetimes and interval censored illness times.

**Value**

A latent variable model object `lvm`

**Author(s)**

Thomas Alexander Gerds
Examples

```
## Not run:
library(lava)
library(prodlim)
# generate illness-death model based on exponentially
# distributed times
m <- idmModel(scale.illtime=1/70,
    shape.illtime=1.8,
    scale.lifetime=1/50,
    shape.lifetime=0.7,
    scale.waittime=1/30,
    shape.waittime=0.7)
round(sim(m,6),1)

# Estimate the parameters of the Weibull models
# based on the uncensored exact event times
# and the uncensored illstatus.
set.seed(18)
d <- sim(m,100,latent=FALSE)
d$uncensored.status <- 1
f <- idm(formula01=Hist(time=illtime,event=illstatus)-1,
    formula02=Hist(time=lifetime,event=uncensored.status)-1,
    data=d,
    conf.int=FALSE)
print(f)

# Change the rate of the 0->2 and 0->1 transitions
# also the rate of the 1->2 transition
# and also lower the censoring rate
m <- idmModel(scale.lifetime=1/2000,
    scale.waittime=1/30,
    scale.illtime=1/1000,
    scale.censtime=1/1000)
set.seed(18)
d <- sim(m,50,latent=TRUE)
d$uncensored.status <- 1
f <- idm(formula01=Hist(time=observed.illtime,event=illstatus)-1,
    formula02=Hist(time=observed.lifetime,event=uncensored.status)-1,
    data=d,
    conf.int=FALSE)
print(f)

# Estimate based on the right censored observations
fc <- idm(formula01=Hist(time=illtime,event=seen.ill)-1,
    formula02=Hist(time=observed.lifetime,event=seen.exit)-1,
    data=d,
    conf.int=FALSE)
print(fc)

# Estimate based on interval censored and right censored observations
fi <- idm(formula01=Hist(time=list(L,R),event=seen.ill)-1,
```
intensity

formula02=Hist(time=observed.lifetime,event=seen.exit)-1,
data=d,
conf.int=FALSE)

print(f1)

# Estimation of covariate effects:
# X1, X2, X3
m <- idmModel(shape.waittime=2,
scale.lifetime=1/2000,
scale.waittime=1/300,
scale.illtime=1/10000,
scale.censtime=1/10000)
distribution(m,"X1") <- binomial.lvm(p=0.3)
distribution(m,"X2") <- normal.lvm(mean=120,sd=20)
distribution(m,"X3") <- normal.lvm(mean=50,sd=20)
regression(m,from="latent.illtime",to="X1") <- 1.7
regression(m,from="latent.illtime",to="X2") <- 0.07
regression(m,from="latent.illtime",to="X3") <- -0.1
regression(m,from="latent.waittime",to="X1") <- 1.8
regression(m,from="latent.lifetime",to="X1") <- 0.7
set.seed(28)
d <- sim(m,100,latent=TRUE)
head(d)
table(ill=d$seen.ill,death=d$seen.exit)

# Estimation based on uncensored data
d$uncensored.status <- 1
# uncensored data
F1 <- idm(formula01=Hist(time=illtime,event=illstatus)-X1+X2+X3,
          formula02=Hist(time=lifetime,event=uncensored.status)-X1+X2+X3,
data=d,conf.int=FALSE)
print(F1)

# Estimation based on right censored data
F2 <- idm(formula01=Hist(time=illtime,event=seen.ill)-X1+X2+X3,
          formula02=Hist(time=observed.lifetime,event=seen.exit)-X1+X2+X3,
data=d,conf.int=FALSE)
print(F2)

# Estimation based on interval censored and right censored data
F3 <- idm(formula01=Hist(time=list(L,R),event=seen.ill)-X1+X2+X3,
          formula02=Hist(time=observed.lifetime,event=seen.exit)-X1+X2+X3,
data=d,conf.int=FALSE)
print(F3)
cbind(uncensored=F1$coef,right.censored=F2$coef,interval.censored=F3$coef)

## End(Not run)

intensity

M-spline estimate of the transition intensity function
Description
M-spline estimate of the transition intensity function and the cumulative transition intensity function for survival and illness-death models

Usage
intensity(times, knots, number.knots, theta, linear.predictor = 0)

Arguments
times Time points at which to estimate the intensity function
knots Knots for the M-spline
number.knots Number of knots for the M-splines (and I-splines see details)
theta The coefficients for the linear combination of M-splines (and I-splines see details)
linear.predictor Linear predictor beta*Z. When it is non-zero, transition and cumulative transition are multiplied by exp(linear.predictor). Default is zero.

Details
The estimate of the transition intensity function is a linear combination of M-splines and the estimate of the cumulative transition intensity function is a linear combination of I-splines (the integral of a M-spline is called I-spline). The coefficients theta are the same for the M-splines and I-splines.
Important: the theta parameters returned by idm and shr are in fact the square root of the splines coefficients. See examples.
This function is a R-translation of a corresponding Fortran function called susp. susp is used internally by idm and shr.

Value
times The time points at which the following estimates are evaluated.
intensity The transition intensity function evaluated at times.
cumulative.intensity The cumulative transition intensity function evaluated at times
survival The "survival" function, i.e., exp(-cumulative.intensity)

Author(s)
R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> and Thomas Alexander Gerds <tag@biostat.ku.dk>

See Also
shr, idm
Examples

data(testdata)
fit.su <- shr(Hist(time=list(l, r), id) ~ cov,
data = testdata, method = "Splines", CV = TRUE)
intensity(times = fit.su$time, knots = fit.su$knots,
number.knots = fit.su$knots, theta = fit.su$theta^2)

## Not run:
data(Paq1000)
fit.idm <- idm(formula02 = Hist(time = t, event = death, entry = e) ~ certif,
formula01 = Hist(time = list(l, r), event = dementia) ~ certif,
formula12 = ~ certif, method = "Splines", data = Paq1000)
# Probability of survival in state 0 at age 80 for a subject with no cep given
# that he is in state 0 at 70
su0 <- (intensity(times = 80, knots = fit.idm$knots01,
number.knots = fit.idm$knots01,
theta = fit.idm$theta01^2)$survival
* intensity(times = 80, knots = fit.idm$knots02,
number.knots = fit.idm$knots02,
theta = fit.idm$theta02^2)$survival)
(intensity(times = 70, knots = fit.idm$knots01,
number.knots = fit.idm$knots01,
theta = fit.idm$theta01^2)$survival
* intensity(times = 70, knots = fit.idm$knots02,
number.knots = fit.idm$knots02,
theta = fit.idm$theta02^2)$survival)
# Same result as:
predict(fit.idm, s = 70, t = 80, conf.int = FALSE) # see first element

## End(Not run)

---

Paq1000 Paquid data set

Description

Paquid data set composed of 1000 subjects selected randomly from the Paquid data set of 3675 subjects.

Format

A data frame with 1000 rows and the following 8 columns.

- **dementia** dementia status, 0=non-demented, 1=demented
- **death** death status, 0=alive, 1=dead
- **e** age at entry in the study
- **l** for demented subjects: age at the visit before the diagnostic visit; for non-demented subjects: age at the last visit (censoring age)
Plot method for an illness-death model

Plot estimated baseline transition intensities from an object of class `idm` optionally with confidence limits.

Usage

```r
plot(x, conf.int = FALSE, citype = "shadow", add = FALSE,
     axes = TRUE, col, lwd, lty, xlim, ylim, xlab, ylab, legend = TRUE,
     transition = c("01", "02", "12"), ...)
```

Arguments

- `x`: a `idmWeib` class object (output from calling `idm` with the (default) option `intensities="Weib"`).
- `conf.int`: If TRUE show confidence limits
- `citype`: Type of confidence limits, can be "shadow" or "bars"
- `add`: If TRUE add to existing plot
- `axes`: If TRUE axes are drawn
- `col`: Color of the lines
- `lwd`: Width of the lines
- `lty`: Type of the lines
- `xlim`: Limits for x-axis
- `ylim`: Limits for y-axis
- `xlab`: Label for x-axis
- `ylab`: Label for y-axis
- `legend`: If TRUE a legend is drawn, which can be further controlled via `SmartControl`
- `transition`: Choose one of the transition intensities: `c("01", "02","12")`
- `...`: Passed to `SmartControl`
Value

Print a plot of the baseline transition intensities of an illness-death model estimated using a Weibull approach.

See Also

print.idm, summary.idm, idm

Examples

library(lava)
library(prodlim)
m <- idmModel(scale.lifetime=1/10, scale.illtime=1/8)
distribution(m, "X") <- binomial.lvm()
regression(m, latent.lifetime="X") <- 0.7
set.seed(30)
d <- sim(m, 100)
fit.weib <- idm(formula02=Hist(observed.lifetime, event=seen.exit)-1,
formula01=Hist(time=list(L,R), event=seen.ill)-1, data=d, conf.int=FALSE)
plot(fit.weib)

## Not run:
## FIXME: the limits for the 01 transition are a bit wide!?  
## with bootstrap confidence limits
fit.weib <- idm(formula02=Hist(observed.lifetime, event=seen.exit)-1,
formula01=Hist(time=list(L,R), event=seen.ill)-1, data=d, conf.int=TRUE)
plot(fit.weib)

## End(Not run)
Arguments

- **x**: A `shrWeib` or a `shrSplines` class object (output from calling `shr` function).
- **type**: Type of function to plot. The default is "shr".
- **add**: Boolean.
- **newdata**: New data.
- **cause**: Cause.
- **col**: Color.
- **lty**: Line type.
- **lwd**: Line width.
- **ylim**: Y limits.
- **xlim**: X limits.
- **xlab**: X label.
- **ylab**: Y label.
- **legend**: Legend.
- **confint**: Confidence interval.
- **timeOrigin**: Time origin.
- **axes**: Axes.
- **percent**: Percent.
- **...**: Other graphical parameters.

Value

Print a plot of a survival model.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

- `plot.shr`

Examples

```r
# Weibull survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)-cov, data=testdata)

# Pointwise confidence limits
plot(fit.su)
```
# no pointwise confidence limits
plot(fit.su, confint=FALSE)

---

**predict.idm**

Predictions for an illness-death model using either a penalized likelihood approach or a Weibull parametrization.

---

**Description**

Predict transition probabilities and cumulative probabilities from an object of class `idmSplines` with confidence intervals are calculated.

**Usage**

```r
## S3 method for class 'idm'
predict(object, s, t, newdata, nsim = 200, seed = 21,
    conf.int = 0.95, lifeExpect = FALSE, maxtime, …)
```

**Arguments**

- `object`: an `idm` class objects returned by a call to the `idm` function
- `s`: time point at which prediction is made.
- `t`: time horizon for prediction.
- `newdata`: A data frame with covariate values for prediction.
- `nsim`: number of simulations for the confidence intervals calculations. The default is 200.
- `seed`: Seed passed to `set.seed` for Monte Carlo simulation of confidence intervals.
- `conf.int`: Level of confidence, i.e., a value between 0 and 1, the default is 0.95. The default is also used when `conf.int=TRUE`. To avoid computation of confidence intervals, set `conf.int` to FALSE or NULL.
- `lifeExpect`: Logical. If TRUE compute life expectancies, i.e., `t=Inf`.
- `maxtime`: The upper limit of integration for calculations of life expectancies from Weibull parametrizations.
- `…`: other parameters.

**Value**

a list containing the following predictions with pointwise confidence intervals:

- `p00`: the transition probability $p_{00}$.
- `p01`: the transition probability $p_{01}$.
- `p11`: the transition probability $p_{11}$.
predict.idm

\[ p_{12} \] the transition probability \( p_{12} \).
\[ p_{02_0} \] the probability of direct transition from state 0 to state 2.
\[ p_{02_1} \] the probability of transition from state 0 to state 2 via state 1.
\[ p_{02} \] transition probability \( p_{02} \). Note that \( p_{02} = p_{02_0} + p_{02_1} \).
\[ F_{01} \] the lifetime risk of disease. \( F_{01} = p_{01} + p_{02_1} \).
\[ F_0 \] the probability of exit from state 0. \( F_0 = p_{02_0} + p_{01} + p_{02_1} \).

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> and Thomas Alexander Gerds <tag@biostat.ku.dk>

See Also

idm

Examples

```r
## Not run:
set.seed(100)
d=simulateIDM(n = 100)
fit <- idm(formula01=Hist(time=list(l,R),event=seen.ill)-X1+X2+X3,
           formula02=Hist(time=observed.lifetime,event=seen.exit)-X1+X2+X3,
           data=d,conf.int=FALSE)
predict(fit,s=0,t=80,conf.int=FALSE)
predict(fit,s=0,t=80,nsim=4,conf.int=TRUE)
predict(fit,s=0,t=80,nsim=4,conf.int=FALSE,lifeExpect=TRUE)

data(Paq1000)
library(prodlim)
fit.paq <- idm(formula02=Hist(time=t,event=death,entry=e)-certif,
               formula01=Hist(time=list(l,r),event=dementia)-certif,
               data=Paq1000)
predict(fit.paq,s=70,t=80,newdata=data.frame(certif=1))
predict(fit.paq,s=70,lifeExpect=TRUE,newdata=data.frame(certif=1))

fit.splines <- idm(formula02=Hist(time=t,event=death,entry=e)-certif,
                   formula01=Hist(time=list(l,r),event=dementia)-certif,
                   formula12=-1,
                   method="Splines",
                   data=Paq1000)
predict(fit.splines,s=70,t=80,newdata=data.frame(certif=1))
predict(fit.splines,s=70,t=80,lifeExpect=TRUE,newdata=data.frame(certif=1),nsim=20)

## End(Not run)
```
Description

Print a summary of a fitted illness-death model

Usage

```r
## S3 method for class 'idm'
print(x, conf.int = 0.95, digits = 4, pvalDigits = 4,
     eps = 0.0001, ...)
```

Arguments

- `x` Class idm object, i.e. the result of a call to the `idm` function with `intensities="Weib"`.
- `conf.int` The level of confidence for the hazard ratios. The default is 0.95.
- `digits` Number of digits to print.
- `pvalDigits` Number of digits to print for p-values.
- `eps` Passed to `format.pval`.
- `...` Not used.

Author(s)

Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr>, Thomas A. Gerds <tag@biostat.ku.dk>

See Also

`summary.idm`, `plot.idm`

Examples

```r
## Not run:
data(Paq1000)
library(prodlim)
fit.splines <- idm(formula02=Hist(time=t,event=death,entry=t0)-certif,
  formula01=Hist(time=list(l,r),event=dementia)-certif,
  formula12=-1,
  method="Splines",
  data=Paq1000)
print(fit.splines)

## End(Not run)
```
Description

Print a summary of a fitted illness-death model using the penalized likelihood approach.

Usage

```r
## S3 method for class 'shr'
print(x, conf.int = 0.95, digits = 4, pvalDigits = 4,
      eps = 0.0001, ...)
```

Arguments

- `x`: a shr object, i.e., the result of a call to the `shr` function with `hazard="Splines"`.
- `conf.int`: The level of confidence for the hazard ratios. The default is 0.95.
- `digits`: number of digits to print.
- `pvalDigits`: number of digits to print for p-values.
- `eps`: convergence criterion used for p-values.
- `...`: other unused arguments.

Author(s)

- R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr>

See Also

- `summary.shr`, `plot.shr`

Examples

```r
## Not run:
# a penalized survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)-cov,data=testdata,method="Splines")
print(fit.su)

## End(Not run)
```
**shr**

*Fit a survival model*

**Description**

Fit a survival model using either a semi-parametric approach (penalized likelihood with an approximation of the hazard function by linear combination of M-splines) or a parametric approach (specifying a Weibull distribution on the hazard function). Left-truncated, right-censored, and interval-censored data are allowed.

**Usage**

```r
shr(formula, data, eps = c(5, 5, 3), n.knots = 7, knots = "equidistant", 
    CV = FALSE, kappa = 10000, conf.int = 0.95, maxiter = 200, 
    method = "Weib", print.iter = FALSE, na.action = na.omit)
```

**Arguments**

- **formula**: a formula object with the response on the left of a `~` operator, and the terms on the right. The response must be a survival object or Hist object as returned by the 'Surv' or 'Hist' function.
- **data**: a data frame in which to interpret the variables named in the formula.
- **eps**: a vector of length 3 for the convergence criteria (criterion for parameters, criterion for likelihood, criterion for second derivatives). The default is `c(5, 5, 3)` and corresponds to criteria equals to $10^{-5}$, $10^{-5}$ and $10^{-3}$.
- **n.knots**: Argument only active for the penalized likelihood approach method="splines". Number of knots for the splines to use to approximate the hazard function. The default is 7. If knots are given as a vector this argument is ignored. The algorithm needs least 5 knots and at most 20 knots.
- **knots**: Argument only active for the penalized likelihood approach method="splines". There are three ways to control the placement of the knots between the smallest and the largest of all time points:
  - knots="equidistant" Knots are placed with same distance on the time scale.
  - knots="quantiles" Knots are placed such that the number of observations is roughly the same between knots.
  - knots=list() List of length 3. The list elements are the actual placements (timepoints) of the knots for the M-spline.

The algorithm needs at least 5 knots and allows no more than 20 knots.
- **CV**: binary variable equals to 1 when search (by approximated cross validation) of the smoothing parameter kappa and 0 otherwise. Argument for the penalized likelihood approach. The default is 0.
- **kappa**: Argument only active for the penalized likelihood approach method="splines". A positive number (smoothing parameter) If CV=1 the value is used as a starting value for a cross validation search to optimize kappa.
conf.int  Level of confidence pointwise confidence intervals of the survival and hazard functions, i.e., a value between 0 and 1, the default is 0.95. The default is also used when conf.int=TRUE. To avoid computation of confidence intervals, set conf.int to FALSE or NULL.

maxiter  maximum number of iterations. The default is 200.

method  type of estimation method: "Splines" for a penalized likelihood approach with approximation of the hazard function by M-splines, "Weib" for a parametric approach with a Weibull distribution on the hazard function. Default is "Weib".

print.iter  boolean parameter. Equals to TRUE to print the likelihood during the iteration process, FALSE otherwise. Default is FALSE. This option is not running on Windows.

na.action  how NAs are treated. The default is first, any na.action attribute of data, second a na.action setting of options, and third 'na.fail' if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL.

Details
The estimated parameters are obtained using the robust Marquardt algorithm (Marquardt, 1963) which is a combination between a Newton-Raphson algorithm and a steepest descent algorithm.

Value

call  regression parameters.
coef  vector containing the log-likelihood without and with covariate.
loglik  vector containing the log-likelihood without and with covariate.
modelPar  Weibull parameters.
N  number of subjects.
NC  number of covariates.
events  number of events.
modelResponse  model response: Hist or Surv object.
converged  integer equal to 1 when the model converged, 2, 3 or 4 otherwise.
time  times for which survival and hazard functions have been evaluated for plotting.
hazard  matched values of the hazard function.
lowerHazard  lower confidence limits for hazard function.
upperHazard  upper confidence limits for hazard function.
surv  matched values of the survival function.
lowerSurv  lower confidence limits for survival function.
upperSurv  upper confidence limits for survival function.
RR  vector of relative risks.
V  variance-covariance matrix.
se  standard errors.
knots  knots of the M-splines estimate of the hazard function.
nknots number of knots.

CV a binary variable equals to 1 when search of the smoothing parameter kappa by approximated cross-validation, 1 otherwise. The default is 0.

niter number of iterations.

cv vector containing the convergence criteria.

na.action observations deleted if missing values.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

References


See Also

shr, print.shr, summary.shr, print.shr,

Examples

```r
# Weibull survival model
library(prodlim)
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id~cov,data=testdata)
fit.su
summary(fit.su)
## Not run:
shr.spline <- shr(Hist(time=list(l,r),id~cov,data=testdata,method="splines",n.knots=6)
shr.spline
shr.spline.q <- shr(Hist(time=list(l,r),id~cov,data=testdata,
                   method="splines",n.knots=6,knots="quantiles")
plot(shr.spline.q)

## manual placement of knots
shr.spline.man <- shr(Hist(time=list(l,r),id~cov,data=testdata,method="splines",knots=seq(0,7,1))
## End(Not run)
```
sim.idmModel          Simulate illness-death model data

Description

Function to simulate illness-death model data

Usage

## S3 method for class 'idmModel'
sim(x, n, illness.known.at.death = TRUE, compliance = 1,
   latent = FALSE, keep.inspectiontimes = FALSE, ...)

Arguments

x              An idmModel object as obtained with idmModel
n              Number of observations
illness.known.at.death
   Affects the value of variable seen.ill
compliance    Probability of missing an inspection time.
latent        if TRUE keep the latent event times
keep.inspectiontimes
   if TRUE keep the inspection times.
...            Extra arguments given to sim

Details

Based on the functionality of the lava PACKAGE

Value

A data set with interval censored observations from an illness-death model

Author(s)

Thomas Alexander Gerds

Examples

eexample(idmModel)
help(idmModel)
**sim.survIC** |  
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simulate interval censored survival data</strong></td>
</tr>
</tbody>
</table>

---

**Description**

Function to simulate interval censored survival data

**Usage**

```r
## S3 method for class 'survIC'
sim(x, n, compliance = 1, latent = TRUE,
   keep.inspectiontimes = FALSE, ...)
```

**Arguments**

- `x`: An survIC object as obtained with survIC
- `n`: Number of observations
- `compliance`: Probability of missing an inspection time.
- `latent`: if TRUE keep the latent event times
- `keep.inspectiontimes`: if TRUE keep the inspection times.
- `...`: Extra arguments given to sim

**Details**

Based on the functionality of the lava PACKAGE

**Value**

A data set with interval censored observations

**Author(s)**

Thomas Alexander Gerds

**Examples**

```r
library(lava)
example(survIC)
help(survIC)
ol <- survIC()
dat.ol <- sim(ol, 10)
```
**simulateIDM**  
Sample illness-death model data

**Description**

Simulate data from an illness-death model with interval censored event times and covariates

**Usage**

`simulateIDM(n = 100)`

**Arguments**

- `n`: number of observations

**Details**

Simulate data from an illness-death model with interval censored event times and covariates for the purpose of illustrating the help pages of the SmoothHazard package. See the body of the function for details, i.e., evaluate `simulateIDM`

**See Also**

`idmModel` `sim.idmModel`

**Examples**

```r
# simulateIDM
simulateIDM(100)
```

**summary.idm**  
Summary of a fitted illness-death model

**Description**

Summarize the event history data of an illness-death regression model and show regression coefficients for transition intensities

**Usage**

```r
## S3 method for class 'idm'
summary(object, conf.int = 0.95, digits = 4, pvalDigits = 4,
         eps = 0.0001, ...)
```
Arguments

object
conf.int
digits
pvalDigits
eps
...  

a idmSplines object, i.e., the result of a call to the idm function with intensities="Splines".
The level of confidence for the hazard ratios. The default is 0.95.
number of digits to print.
number of digits to print for p-values.
convergence criterion used for p-values.
other unused arguments.

Author(s)

R: Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr> Fortran: Pierre Joly <Pierre.Joly@isped.u-bordeaux2.fr>

See Also

idm, print.idm, plot.idm

Examples

## Not run:
library(prodlim)
data(Paq1000)
fit.splines <- idm(formula02=Hist(time=t,event=death,entry=e)-certif,
                   formula01=Hist(time=list(1,r),event=dementia)-certif,
                   formula12=-1,
                   method="Splines",
data=Paq1000)
summary(fit.splines)

## End(Not run)

summary.shr

Summary of a fitted survival model using a penalized likelihood approach

Description

Print a short summary of a fitted illness-death model using the penalized likelihood approach.

Usage

## S3 method for class 'shr'
summary(object, conf.int = 0.95, digits = 4, pvalDigits = 4,
         eps = 0.0001, ...)
Arguments

 object a shr object, i.e., the result of a call to the shr function.
 conf.int The level of confidence for the hazard ratios. The default is 0.95.
 digits number of digits to print.
 pvalDigits number of digits to print for p-values.
 eps convergence criterion used for p-values.
 ... other unused arguments.

Author(s)

Celia Touraine <Celia.Touraine@isped.u-bordeaux2.fr>

See Also

shr, print.shr, plot.shr

Examples

## Not run:
# a penalized survival model
data(testdata)
library(prodlim)
fit.su <- shr(Hist(time=list(l,r),id)-cov,data=testdata,method="Splines")
summary(fit.su)

# Weibull survival model
data(testdata)
fit.su <- shr(Hist(time=list(l,r),id)-cov,data=testdata)
summary(fit.su)

## End(Not run)

---

survIC Generate survival model objects

Description

Function to generate a latent variable model for interval censored survival times.

Usage

survIC(scale.time = 1/100, shape.time = 1, n.inspections = 5, schedule = 10, punctuality = 5)
Arguments

- `scale.time`: Weibull scale for latent time
- `shape.time`: Weibull shape for latent time
- `n.inspections`: Number of inspection times
- `schedule`: Mean of the waiting time between adjacent inspections.
- `punctuality`: Standard deviation of waiting time between inspections.

Details

Based on the functionality of the lava PACKAGE the function generates a latent variable model with a latent time and a censoring mechanism (censtime, inspection1, inspection2,...,inspectionK).

The function `sim.survIC` then simulates interval censored times.

Value

A latent variable model object `lvm`

Author(s)

Thomas Alexander Gerds

Examples

```r
## Not run:
library(lava)
library(prodlim)
# generate survival model based on exponentially distributed times
m <- survIC(scale.time=1/50, shape.time=0.7)
round(sim(m,6),1)

# Estimate the parameters of the Weibull models
# based on the uncensored exact event times
# and the uncensored illstatus.
set.seed(18)
d <- sim(m,100,latent=FALSE)
d$uncensored.status <- 1
f <- shr(Hist(time=list(L,R),event=uncensored.status)-1,
data=d,
        conf.int=FALSE)
print(f)
## End(Not run)
```
testimonial

**Description**

A simulated data frame for survival models composed of right-censored and interval-censored data.

**Format**

A data frame with 936 observations on the following 4 variables.

- **l** for diseased subjects: left endpoint of censoring interval; for non-diseased subjects: right censoring time
- **r** for diseased subjects: right endpoint of censoring interval; for non-diseased subjects: right censoring time for the disease event
- **id** disease status
- **cov** covariate

**Examples**

data(testdata)
head(testdata)
Index

*Topic datasets
  Paq1000, 11
testdata, 28

*Topic illness-death
  idm, 2

*Topic methods
  plot.idm, 12
  plot.shr, 13
  predict.idm, 15
  print.idm, 17
  print.shr, 18
  shr, 19
  summary.idm, 24
  summary.shr, 25

idm, 2, 10, 13, 15–17, 25
idmModel, 7
intensity, 9

kappa, 5, 21

Paq1000, 11
plot.idm, 12, 17, 25
plot.shr, 13, 14, 18, 26
predict.idm, 5, 15
print.idm, 5, 13, 17, 25
print.shr, 18, 21, 26

shr, 10, 14, 18, 19, 21, 26
sim.idmModel, 7, 22
sim.survIC, 23, 27
simulateIDM, 24
SmartControl, 12
summary.idm, 5, 13, 17, 24
summary.shr, 18, 21, 25
survIC, 26
survModelIC(survIC), 26
testdata, 28