Package ‘tvcure’

April 7, 2025

Type Package

Title Additive Cure Survival Model with Time-Varying Covariates

Version 0.6.6

Maintainer Philippe Lambert <`p.lambert@uliege.be`>

BugReports `https://github.com/plambertULiege/tvcure/issues`

Description Fit of a double additive cure survival model with time-varying covariates. The additive terms in the long- and short-term survival submodels, modelling the cure probability and the event timing for susceptible units, are estimated using Laplace P-splines. For more details, see Lambert and Kreyenfeld (2025) <doi:10.1093/jrsssa/qnaf035>.

License GPL-3

URL <`https://github.com/plambertULiege/tvcure`>

Encoding UTF-8

Imports cubicBsplines, Rfast, MASS, Matrix, mgcv

RoxygenNote 7.3.2

NeedsCompilation no

Author Philippe Lambert [aut, cre] (<`https://orcid.org/0000-0002-3670-3328`>)

Repository CRAN

Date/Publication 2025-04-07 16:00:01 UTC

Contents

`additive.tvcure` *Extract additive term estimates from a tvcure object.*

Description

Extract additive term estimates from a tvcure object.

Usage

```
additive.tvcure(obj.tvcure, ngrid = 300, ci.level = 0.95)
```

Arguments

`obj.tvcure`	a `tvcure.object`.
`ngrid`	number of gridpoints where the fitted additive terms are evaluated.
`ci.level`	confidence level for the pointwise credible intervals of the additive terms.

Value

A list with following elements:

- `f0` : a function estimate of f_0.
- `F0` : a function estimate of F_0.
- `T` : the follow-up time after which a unit is considered ‘cured’.
- `nfixed1` : the number of non-penalized regression parameter in the long-term term (or quantum) submodel.
- `J1` : number of additive terms in the long-term term (or quantum) submodel.
- `additive.lab1` : labels of the additive terms in the long-term term (or quantum) submodel.
- `K1` : number of P-spline parameters per additive term in the long-term term (or quantum) submodel.
- `knots1` : list of length J1 containing the knots of the additive term in the long-term term (or quantum) submodel.

- `f1.grid` : list of length J1 containing for each additive term in the long-term term (or quantum) submodel, a list of length 3 with elements <x>, <y.mat> and <y.mat2>
  - Element <x> is a vector of `ngrid` equidistant values covering the range of values for the covariate ;
  - <y.mat> is (ngrid x 3) matrix containing in column 1 the estimated values of the additive term at <x> and the bounds of the pointwise credible interval for it in the other 2 columns.
  - <y.mat2> is (ngrid x 3) matrix containing in column 1 the estimated values of the additive term at <x> and the bounds of the simultaneous credible region for it in the other 2 columns.
- `f1` : list of length J1 containing the estimated function of the corresponding additive term in the long-term term (or quantum) submodel.
- `f1.se` : list of length J1 containing the estimated standard error function of the corresponding additive term in the long-term term (or quantum) submodel.

The same definitions applies for `nfixed2`, `J2`, `additive.lab2`, `K2`, `knots2`, `f2.grid`, `f2`, `f2.se` with the additive terms in the short-term (or timing) submodel.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)

## Extract additive term estimates from tvcure object
obj = additive.tvcure(model)
names(obj)
```

`AIC.tvcure` *Akaike Information Criterion (AIC) of a tvcure object.*

Description

Akaike Information Criterion (AIC) for the fitted tvcure model in a `tvcure.object`.

Usage

```
## S3 method for class 'tvcure'
AIC(object, ..., k=2)
```

Arguments

`object`	A `tvcure.object`.
`...`	Other optional tvcure objects.
`k`	The penalty per parameter to be used. (Default: k=2 for the classical AIC).

Details

Akaike information criterion for the fitted model in a tvcure object, with a penalty calculated using the total effective degrees of freedom, -2log(L) + 2*ED.tot, smaller values being preferred during model selection.

Value

The AIC as a numeric value, computed according to the model specified in the input `object`.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `BIC.tvcure`, `logEvid`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
AIC(model)
```

`BIC.tvcure` *Bayesian Information Criterion (BIC) of a* `tvcure.object`.

Description

Bayesian Information Criterion (BIC) for the fitted tvcure model in a `tvcure.object`.

Usage

```
## S3 method for class 'tvcure'
BIC(object, ...)
```

Arguments

`object`	An object of class `tvcure.object`.
`...`	Optionally more fitted objects.

Details

Bayesian (Schwarz) information criterion in a tvcure object, with a penalty calculated using the total effective degrees of freedom and the total number of observed events, -2log(L) + log(d)*ED.tot, smaller values being preferred during model selection.

Value

The BIC as a numeric value, computed according to the model specified in the input `object`.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `AIC.tvcure`, `logEvid`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
BIC(model)
```

`centeredBasis.gen` *Generation of a recentered cubic B-spline basis matrix.*

Description

Generation of a cubic B-spline basis matrix with recentered columns to handle the identifiability constraint in additive models. See Wood (CRC Press 2017, pp. 175-176) for more details.

Usage

```
centeredBasis.gen(x, knots, cm = NULL, pen.order = 2)
```

Arguments

`x`	Vector of values where the "recentered" B-spline basis is evaluated.
`knots`	Vector of knots that must cover the values in `x`.
`cm`	(Optional) values subtracted from each column of the original B-spline matrix.
`pen.order`	Order of the penalty applied on B-spline parameters. (Default: 2).

Value

List containing

- `B` : centered cubic B-spline matrix obtained by subtracting `cm[j]` from the jth B-spline in column j of the original B-spline matrix evaluated at `x`.
- `Dd` : difference matrix (of order `pen.order`) for the associated centered B-spline matrix.
- `Pd` : penalty matrix (of order `pen.order`) for the associated centered B-spline matrix.
- `K` : number of centered B-splines in the basis.
- `cm` : values subtracted from each column of the original B-spline matrix. By default, this is a vector containing the mean of each column in the original B-spline matrix.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
x = seq(0,1,by=.01)
knots = seq(0,1,length=5)
obj = centeredBasis.gen(x,knots)
matplot(x,obj$B,type="l",ylab="Centered B-splines")
colMeans(obj$B)
```

`deviance.tvcure` *Deviance of a* `tvcure.object`.

Description

Deviance for the fitted tvcure model in a `tvcure.object`.

Usage

```
## S3 method for class 'tvcure'
deviance(object, ...)
```

Arguments

`object`	An object of class `tvcure.object`.
`...`	Optionally more fitted objects.

Details

Deviance -2log(L/L.hat)

Value

The deviance as a numeric value, computed according to the model specified in the input `object`.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `AIC.tvcure`, `BIC.tvcure`, `logEvid`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
deviance(model)
```

`EDF` *Compute the effective degrees freedom in a tvcure model*

Description

Compute the effective degrees freedom in a tvcure model

Usage

```
EDF(model, Wood.test = FALSE, joint.computation = TRUE)
```

Arguments

`model`	A tvcure object
`Wood.test`	Logical indicating if P-values based on Wood's test (Biometrika 2013) of the significance of additive terms should be preferred over basic Chi-square tests. (Default: FALSE).
`joint.computation`	Logical indicating if variance-covariance matrices for the regression and spline parameters in the long- and short-term survival submodels should be computed jointly (TRUE) or separately (FALSE). (Default: TRUE).

Value

A list containing the effective degrees of freedom for the additive terms in the long-term (quantum) and short-term (timing) survival submodels, with the selected statistical test for significance and its P-value.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
EDF(model)
```

`logEvid` *Generic function for computing log-evidence*

Description

This is a generic function returning the log-evidence for the class of the input object.

Usage

```
logEvid(object, ...)
```

Arguments

`object`	An object for which log-evidence is to be computed.
`...`	Additional arguments (not used for now).

Value

The log-evidence as a numeric value, computed according to the model specified in the input object.

`logEvid.tvcure` *Log-evidence of a tvcure object.*

Description

The log-evidence of the fitted tvcure model in a `tvcure.object`.

Usage

```
## S3 method for class 'tvcure'
logEvid(object, ...)
```

Arguments

`object`	A `tvcure.object`.
`...`	Optionally more tvcure objects.

Details

Provides the log-evidence (or log-marginal likelihood) of the fitted tvcure model in a given `tvcure.object`, where the evidence is the marginal posterior of the penalty parameters at their selected values.

Value

The log-evidence as a numeric value, computed according to the model specified in the input `object`.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `AIC.tvcure`, `BIC.tvcure`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
logEvid(model)
```

`Pcal.fun` *Function to generate a penalty matrix for additive terms.*

Description

Compute the penalty matrix associated to a vector containing fixed (non-penalized) parameters and equal-size sub-vectors of penalized spline parameters.

Usage

```
Pcal.fun(nfixed, lambda, Pd.x)
```

Arguments

`nfixed`	the number of fixed (i.e. non-penalized) parameters.
`lambda`	a vector of `p` penalty parameters where each component is associated to a sub-vector of spline parameters of length `J`.
`Pd.x`	a penalty matrix of size `J` associated to a given sub-vector of spline parameters. .

Value

A block diagonal penalty matrix of size (`nfixed+pJ`) given by Blockdiag(diag(0,`nfixed`), diag(`lambda`).kron.`Pd.x`).

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
Dd = diff(diag(1,5),diff=2) ## Difference penalty matrix for a vector of length 5
Pd = t(Dd) %*% Dd ## Penalty matrix of order 2
nfixed = 2 ## 2 unpenalized parameters
## Global penalty matrix when 2 unpenalized parameters and 2 additive terms with
##   2 vectors of 5 P-splines coefficients with lambda values 10 and 100 respectively.
Pcal.fun(nfixed=2,lambda=c(10,100),Pd)
```

`plot.tvcure` *Plot visual information related to a tvcure object.*

Description

Visualization of the estimated additive terms and of the reference (cumulative) hazard function in a tvcure object.

Usage

```
## S3 method for class 'tvcure'
plot(x, ngrid=300, ci.level=.95, pages=0, select=NULL,
                                fill=TRUE, pointwise=TRUE, mar=c(4,5,1,1),
                                xlim0=NULL, ylim0=NULL,
                                xlim1=NULL, ylim1=NULL, xlim2=NULL, ylim2=NULL,
                                equal.ylims=TRUE,...)
```

Arguments

`x`	a `tvcure.object`.
`ngrid`	(optional) number of points used to plot the fitted additive terms. (Default: 300).
`ci.level`	(optional) nominal level for the plotted pointwise credible intervals. (Default: 0.95).
`pages`	The number of pages over which to spread the output. For example, if pages=1 then all terms will be plotted on one page with the layout performed automatically. Set to 0 to have the routine leave all graphics settings as they are. (Default 0).
`select`	Allows the plot for a single model term to be selected for printing. e.g. if you just want the plot for the second smooth term set select=2. The plot of the reference hazard $f_0(t)$ and cumulative hazard $F_0(t)$ functions is provided when select=0. When select=-1, only the reference hazard is plotted.(Default: NULL).

`fill`	Logical indicating whether credible regions should be greyed out. (Default: TRUE).
`pointwise`	Logical indicating whether only pointwise credible intervals should be plotted for additive terms. When FALSE, simultaneous credible regions are also provided. (Default: TRUE).
`mar`	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. (Default: c(4,5,1,1)).
`xlim0`	Vector of length 2 specifying x-axis limits when plotting the estimated reference hazard function $\exp(\beta_0) f_0(t)$.
`ylim0`	Vector of length 2 specifying y-axis limits when plotting the estimated reference hazard function $\exp(\beta_0) f_0(t)$.
`xlim1`	Vector of length 2 specifying (common) x-axis limits when plotting the fitted additive term(s) in the long-term survival submodel. (Default: NULL).
`ylim1`	Vector of length 2 specifying (common) y-axis limits when plotting the fitted additive term(s) in the long-term survival submodel. (Default: NULL).
`xlim2`	Vector of length 2 specifying (common) x-axis limits when plotting the fitted additive term(s) in the short-term survival submodel. (Default: NULL).
`ylim2`	Vector of length 2 specifying (common) y-axis limits when plotting the fitted additive term(s) in the short-term survival submodel. (Default: NULL).
`equal.ylims`	logical indicating if the same y-limits must be used when plotting the fitted additive terms from the same submodel. It can be overriden by non-NULL values for `ylim1` or `ylim2`. (Default: TRUE).
`...`	additional generic plotting arguments.

Details

Plot of the fitted additive terms, as well as of the reference hazard $f_0(t)$ and cumulative hazard $F_0(t)$ functions of the fitted tvcure model in `x`.

Value

In addition to the plots, an invisible list generated by the `additive.tvcure` function is returned.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`> based on the plot.gam function in mgcv for the `pages` argument.

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `print.tvcure`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
plot(model,pages=1)
```

`plotRegion` *Generic function to plot a shaded region around values of a scalar function.*

Description

Generic function to plot a shaded region around values of a scalar function.

Usage

```
plotRegion(x, mat,
            add=FALSE,xlim=range(x),ylim=range(mat),
            colfill="#D9D9D980",lwd=2,xlab="",ylab="",main="",...)
```

Arguments

`x`	n-vector with a grid of x values where the scalar function f(x) is evaluated.
`mat`	(n x 3)-matrix containing on its ith row, the function value at x[i] and the bounds of an interval containing it, (f(x[i]), f.low(x[i]), f.up(x[i])).
`add`	logical indicating if the shaded region should be superposed to an existing plot.
`xlim`	x-limits. (Default: range of x).
`ylim`	y-limits. (Default: range of mat).
`colfill`	color used for filling the shaded region. (Default: "#D9D9D980").
`lwd`	line width to plot (x,f(x)). (Default: 2).
`xlab`	x-label. (Default: none).
`ylab`	y-label. (Default: none).
`main`	plot main title. (Default: none)
`...`	additional generic plotting arguments.

Value

No returned value (in addition to the plot).

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
obj = additive.tvcure(model) ## Extract additive terms

## Plot some of the fitted additive terms
## par(mfrow=c(1,2))
with(obj$f1.grid$x1, plotRegion(x=x,mat=y.mat,xlab="x1",ylab="f(x1)"))
with(obj$f1.grid$x2, plotRegion(x=x,mat=y.mat,xlab="x2",ylab="f(x2)"))
```

`predict.tvcure` *Predict method for tvcure model fits*

Description

Predicted values based on a tvcure object.

Usage

```
## S3 method for class 'tvcure'
predict(object, newdata, ci.level=.95, ...)
```

Arguments

`object`	A `tvcure.object`.
`newdata`	A data frame in which to look for the 'id' (distinguishing the different units), 'time' and covariate values for which 'predictions' should be made. Time values for a given 'id' should be a series of consecutive integers starting with 1. If `newdata$id` does not exist, then predictions are assumed to concern a single unit with consecutive time values starting with 1.
`ci.level`	Credible level for the reported estimates. (Default: 0.95).
`...`	additional generic arguments.

Value

A data frame containing, in addition to the optional `newdata` entries, the following elements:

- `Hp` : Matrix containing estimates of the cumulative population hazard $H_p(t|x_{1:t})$ with its credible interval bounds at time t given the history of covariates.
- `lHp` : Matrix containing estimates of the log cumulative population hazard $\log H_p(t|x_{1:t})$ with its standard error and credible interval bounds at time t given the history of covariates.
- `se.lHp` : Vector containing the standard errors of the estimated log cumulative population hazard at time t given the history of covariates.
- `hp` : Matrix containing estimates of the population hazard $h_p(t|x_{1:t})$ with its credible interval bounds at time t given the history of covariates.
- `lhp` : Matrix containing estimates of the log population hazard $\log h_p(t|x_{1:t})$ with its standard error and credible interval bounds at time t given the history of covariates.
- `se.lhp` : Vector containing the standard errors of the estimated log population hazard at time t given the history of covariates.
- `Sp` : Matrix containing estimates of the population survival fuction $S_p(t|x_{1:t}) = \exp(-H_p(t|x_{1:t}))$ with its credible interval bounds at time t given the history of covariates.
- `pcure` : Matrix containing estimates of the conditional cure probability of a unit still at tisk at time t, $P(T = +\infty|T > t, x = x_t)$, with its credible interval bounds at time t if covariates remain constant from time t.
- `llpcure` : Matrix containing estimates of the conditional log-log cure probability of a unit still at tisk at time t, $\log(-\log P(T = +\infty|T > t, x = x_t))$, with its standard error and credible interval bounds at time t if covariates remain constant from time t.
- `se.llpcure` : Vector containing the standard errors of the estimated conditional log-log cure probability of a unit still at tisk at time t, $\log(-\log P(T = +\infty|T > t, x = x_t))$, if covariates remain constant from time t.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)

## Covariate profiles for which 'predicted' values are requested
newdata = subset(data, id==1 | id==4)[,-3] ## Focus on units 1 & 4
pred = predict(model,newdata)

## Visualize the estimated population survival fns for units 1 & 4
## par(mfrow=c(1,2))
with(subset(pred,id==1), plotRegion(time,Sp,main="Id=1",
                              ylim=c(0,1),xlab="t",ylab="Sp(t)"))
with(subset(pred,id==4), plotRegion(time,Sp,main="Id=4",
                              ylim=c(0,1),xlab="t",ylab="Sp(t)"))
```

`print.tvcure` *Print summary information on a* `tvcure.object`.

Description

Print summary information on a `tvcure.object` generated by `tvcure`.

Usage

```
## S3 method for class 'tvcure'
print(x, ci.level=.95,expEst=TRUE,
             digits.est=3,digits.edf=2,digits.tst=2,digits.Pvalue=3,...)
```

Arguments

`x`	an object of class `tvcure.object`.
`ci.level`	(optional) nominal level for the plotted pointwise credible intervals (default: .95).
`expEst`	(optional) logical indicating if the exponential of the regression parameter estimates should also be provided (with their confidence interval) (default: TRUE).
`digits.est`	number of digits when reporting parameter estimates (default: 3).
`digits.edf`	number of digits when reporting effective degrees of freedom (default: 2).
`digits.tst`	number of digits when reporting test statistics (default: 2).
`digits.Pvalue`	number of digits when reporting P-values (default: 3).
`...`	additional generic printing arguments.

Details

Provides summary measures on the estimation of the regression parameters and additive terms in the tvcure model corresponding to a `tvcure.object` generated by `tvcure`.

Value

No returned value (just printed summary).

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `tvcure.object`, `plot.tvcure`

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
print(model)
```

`qknots` *Specification of the knots of a cubic B-spline basis for given data.*

Description

Specification of the knots of a cubic B-spline basis for given data.

Usage

```
qknots(x, xmin=NULL, xmax=NULL,
       equid.knots = TRUE, pen.order=2, K=25)
```

Arguments

`x`	data that should be supported by the knots of the B-spline basis.
`xmin`	(Optional) minimum value for the knots.
`xmax`	(Optional) maximum value for the knots.
`equid.knots`	Logical indicating if equidistant knots are desired (Default: TRUE).
`pen.order`	penalty order (if equid.knots = TRUE) (Default: 2).
`K`	number of B-splines in the basis (Default: 25).

Value

a list containing the following elements:

- `xmin` : minimum value of the knots.
- `xmax` : maximum value of the knots.
- `knots` : vector containing the knots: equidistant if `equid.knots` is TRUE, based on quantiles of `x` otherwise.
- `Pd` : penalty matrix for the B-spline coefficients.
- `pen.order` : penalty order for the P-spline model.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
x = rnorm(100)
qknots(x)
```

`show_c` *Show Copyright Information*

Description

Displays the copyright information for the package in interactive mode.

Usage

```
show_c()
```

Value

No return value, called for its side effects of printing a message.

`show_w` *Show Warranty Disclaimer*

Description

Displays a notice that the program comes with absolutely no warranty.

Usage

```
show_w()
```

Value

No return value, called for its side effects of printing a message.

`simulateTVcureData` *Simulation of survival data with a cure fraction and time-varying covariates.*

Description

Simulation of cure survival data in a counting process format with time-varying covariates. The population hazard at time t underlying the tvcure model is, for given covariate values,

$$h_p(t|\mathbf{v}(t),\tilde{\mathbf{v}}(t)) = \mathrm{e}^{\eta_\vartheta(\mathbf{v}(t))+\eta_F(\tilde{\mathbf{v}}(t))} f_0(t) S_0(t)^{\exp(\eta_F(\tilde{\mathbf{v}}(t)))-1}$$

with linear predictors

$$\eta_\vartheta(\mathbf{v}(t)) = \beta_0 + \beta_1 z_1(t) + \beta_2 z_2 + f_1(x_1(t)) + f_2(x_2(t))$$
$$\eta_F(\tilde{\mathbf{v}}(t)) = \gamma_1 z_3(t) + \gamma_2 z_4 + \tilde{f}_1(x_3(t)) + \tilde{f}_2(x_4(t))$$

where $\mathbf{v}(t) = (z_1(t), z_2, x_1(t), x_2(t))$, $\tilde{\mathbf{v}}(t) = (z_3(t), z_4, x_3(t), x_4(t))$, with time-varying covariates $x_1(t)$, $x_3(t)$ assumed identical and shared by the 2 submodels when `shared.cov` is TRUE.

The density $f_0(t)$ governing the reference cumulative hazard dynamic is, by default, a Weibull with shape parameter 2.65 and scale parameter 100, ensuring that all susceptible units will experience the monitored event by time Tmax=300.

The functions defining the additive terms are

$$f_1(x_1) = -.63 + .57\arctan(4x_1)\ ;\ f_2(x_2) = -.5\cos(2\pi x_2)$$
$$\tilde{f}_1(\tilde{x}_3) = .15 - .5\cos\big(\pi(\tilde{x}_3 - .75)\big)\ ;\ \tilde{f}_2(\tilde{x}_4) = .8\big(\tilde{x}_4 - .5\big)$$

Covariates are generated as follows:

- $z_1(t), z_3(t)$ are recentered time-varying Bernoulli(0.5) on $(0, T_{max})$;
- $z_2, z_4 \sim N(0,1)$;
- $x_1(t), x_2(t), x_3(t), x_4(t)$ follow random cubic polynomial trajectories on $(0, T_{max})$.

More details can be found in Lambert and Kreyenfeld (2024).

Usage

```
simulateTVcureData(n, seed, Tmax=300,
       f0F0 = list(f0=function(x) dweibull(x, 2.65, 100),
                   F0=function(x) pweibull(x, 2.65, 100)),
       beta, gam, shared.cov=TRUE,
       RC.dist=c("uniform","exponential","Tmax"),
       tRC.min = 120, mu.cens=40, get.details=TRUE)
```

Arguments

`n`	Number of units.
`seed`	Seed (integer) for the random TVcure data generator.
`Tmax`	Maximum follow-up time after which a unit is considered cured in the absence of a previous event. (Default: 300).
`f0F0`	List of length 2 providing the density $f_0(t)$ and associated cdf $F_0(t)$ governing the bounded hazard dynamic on (0,Tmax), with F_0(Tmax)=1.0. (Default: f0F0 = list(f0=function(x) dweibull(x, 2.65, 100), F0=function(x) pweibull(x, 2.65, 100))).
`beta`	3-vector with the regression coefficients <beta> in the long-term (cure) survival (or quantum) submodel.
`gam`	2-vector with the regression coefficients <gamma> in the short-term (cure) survival (or timing) submodel.
`shared.cov`	Logical indicating whether shared covariates for both submodels are assumed, with then $x_1(t) = x_3(t)$. (Default: TRUE).
`RC.dist`	Right-censoring distribution: "uniform" (Uniform on (`tRC.min`,`Tmax`)),"exponential" (with mean `mu.cens`) or "Tmax" (when right-censoring occurs at Tmax)
`tRC.min`	Minimum right-censoring time value if the right-censoring time distribution is Uniform. (Default: 120).
`mu.cens`	Mean of the right-censoring time distribution if it is Exponential. (Default: 40).
`get.details`	Logical indicating if a detailed data frame `rawdata` including the sequence of time-varying covariate values should also be reported. (Default: TRUE).

Value

A list with following elements:

- `seeds` : Seeds used to generate the data for each of the n units.
- `tRC.min` : Minimum right-censoring time value if the right-censoring time distribution is Uniform.
- `RC.dist` : Right-censoring distribution ("Uniform", "Exponential" or "Tmax").
- `cure.rate` : Underlying proportion of cured units (i.e. without an observed event by `Tmax` if the follow-up is not interrupted by that time due to right-censoring).
- `RC.rate` : Observed right-censoring rate.
- `rawdata` : Data frame containing the generated data in a counting process format with the detailed follow-up for each unit until the event or right-censoring occurs:

  - `id` : Unit identificator for each row.
  - `time` : Discrete observation times, starting at 1 for a given unit, until the end of its follow-up. The number of rows associated to a given unit corresponds to the follow-up duration.
  - `event` : Event indicator (1 if it occured, 0 otherwise) for given unit at a given time.
  - `z1, z2, z3, z4, x1, x2, x3, x4` : Covariate values for a given unit at a given time.
- `data.summary` : Data frame with n rows containing summarized information on the generated data for each unit:
  - `id` : Unit identificator (the ith row corresponding to the ith unit).
  - `t.obs` : Observed event or right-censoring time.
  - `delta` : Event indicator (1 if it occured, 0 otherwise).
  - `t.true` : True (possibly unobserved) event time (Inf for a cured unit).
  - `t.cens` : True (possibly unobserved) right-censoring time.
  - `cured` : True (possibly unobserved) cure status.
- `parameters` : List containing the defining elements of the tvcure model:
  - `beta` : The regression parameters in the long-term survival (or quantum) submodel.
  - `gam` : The regression parameters in the short-term survival (or timing) submodel.
  - `f.theta` : A list of length 2 containing the functions defining the additive terms in the long-term survival (or quantum) submodel.
  - `f.gam` : A list of length 2 containing the functions defining the additive terms in the short-term survival (or timing) submodel.
  - `f0` : Density function governing the dynamic of the reference cumulative hazard on (0,Tmax).
  - `F0` : CDF governing the dynamic of the reference cumulative hazard on (0,Tmax).
- `call` : Function call.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Regression parameters
beta = c(beta0=.4, beta1=-.2, beta2=.15) ##  beta0 tunes the cure rate
gam = c(gam1=.2, gam2=.2)
## Data simulation
temp = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)
head(temp$rawdata) ## Overview of the simulated raw data
head(temp$data.summary) ## Overview of the summarized data
```

```
with(temp, c(cure.rate=cure.rate,RC.rate=RC.rate)) ## Cure and RC rates
```

`simulateTVcureData2` *Simulation of survival data with a cure fraction and time-varying covariates.*

Description

Simulation of cure survival data in a counting process format with time-varying covariates. The population hazard at time t underlying the tvcure model is, for given covariate values,

$$h_p(t|\mathbf{v}(t),\tilde{\mathbf{v}}(t)) = \mathrm{e}^{\eta_\vartheta(\mathbf{v}(t))+\eta_F(\tilde{\mathbf{v}}(t))} f_0(t) S_0(t)^{\exp(\eta_F(\tilde{\mathbf{v}}(t)))-1}$$

with linear predictors

$$\eta_\vartheta(\mathbf{v}(t)) = \beta_0 + \beta_1 z_1(t) + \beta_2 z_2 + \beta_{f_1} f_1(x_1(t)) + \beta_{f_2} f_2(x_2(t))$$

$$\eta_F(\tilde{\mathbf{v}}(t)) = \gamma_1 z_3(t) + \gamma_2 z_4 + \gamma_{f_1} \tilde{f}_1(x_3(t)) + \gamma_{f_2} \tilde{f}_2(x_4(t))$$

where $\mathbf{v}(t) = (z_1(t), z_2, x_1(t), x_2(t))$, $\tilde{\mathbf{v}}(t) = (z_3(t), z_4, x_3(t), x_4(t))$, with time-varying covariates $x_1(t), x_3(t)$ assumed identical and shared by the 2 submodels when `shared.cov` is TRUE.

The density $f_0(t)$ governing the reference cumulative hazard dynamic is, by default, a Weibull with shape parameter 2.65 and scale parameter 100, ensuring that all susceptible units will experience the monitored event by time Tmax=300.

The functions defining the additive terms are

$$f_1(x_1) = -.63 + .57\arctan(4x_1)\ ;\ f_2(x_2) = -.5\cos(2\pi x_2)$$

$$\tilde{f}_1(\tilde{x}_3) = .15 - .5\cos\big(\pi(\tilde{x}_3 - .75)\big)\ ;\ \tilde{f}_2(\tilde{x}_4) = .8\big(\tilde{x}_4 - .5\big)$$

Covariates are generated as follows:

- $z_1(t), z_3(t)$ are recentered time-varying Bernoulli(0.5) on $(0, T_{max})$;
- $z_2, z_4 \sim N(0,1)$;
- $x_1(t), x_2(t), x_3(t), x_4(t)$ follow random cubic polynomial trajectories on $(0, T_{max})$.

More details can be found in Lambert and Kreyenfeld (2024).

Usage

```
simulateTVcureData2(n, seed, Tmax=300,
      f0F0 = list(f0=function(x) dweibull(x, 2.65, 100),
                  F0=function(x) pweibull(x, 2.65, 100)),
      beta, gam, beta.f = rep(1,2), gam.f = rep(1,2), shared.cov=TRUE,
      RC.dist=c("uniform","exponential","Tmax"),
      tRC.min = 120, mu.cens=40, get.details=TRUE)
```

Arguments

`n`	Number of units.
`seed`	Seed (integer) for the random TVcure data generator.
`Tmax`	Maximum follow-up time after which a unit is considered cured in the absence of a previous event. (Default: 300).
`f0F0`	List of length 2 providing the density $f_0(t)$ and associated cdf $F_0(t)$ governing the bounded hazard dynamic on (0,Tmax), with F_0(Tmax)=1.0. (Default: f0F0 = list(f0=function(x) dweibull(x, 2.65, 100), F0=function(x) pweibull(x, 2.65, 100))).
`beta`	3-vector with the regression coefficients <beta> in the long-term (cure) survival (or quantum) submodel.
`gam`	2-vector with the regression coefficients <gamma> in the short-term (cure) survival (or timing) submodel.
`beta.f`	2-vector with the multiplying coefficients of the additive terms in long-term (cure) survival (or quantum) submodel.
`gam.f`	2-vector with the multiplying coefficients of the additive terms in the short-term (cure) survival (or timing) submodel.
`shared.cov`	Logical indicating whether shared covariates for both submodels are assumed, with then $x_1(t) = x_3(t)$. (Default: TRUE).
`RC.dist`	Right-censoring distribution: "uniform" (Uniform on (`tRC.min`,`Tmax`)),"exponential" (with mean `mu.cens`) or "Tmax" (when right-censoring occurs at Tmax)
`tRC.min`	Minimum right-censoring time value if the right-censoring time distribution is Uniform. (Default: 120).
`mu.cens`	Mean of the right-censoring time distribution if it is Exponential. (Default: 40).
`get.details`	Logical indicating if a detailed data frame `rawdata` including the sequence of time-varying covariate values should also be reported. (Default: TRUE).

Value

A list with following elements:

- `seeds` : Seeds used to generate the data for each of the n units.
- `tRC.min` : Minimum right-censoring time value if the right-censoring time distribution is Uniform.
- `RC.dist` : Right-censoring distribution ("Uniform", "Exponential" or "Tmax").
- `cure.rate` : Underlying proportion of cured units (i.e. without an observed event by `Tmax` if the follow-up is not interrupted by that time due to right-censoring).
- `RC.rate` : Observed right-censoring rate.
- `rawdata` : Data frame containing the generated data in a counting process format with the detailed follow-up for each unit until the event or right-censoring occurs:
  - `id` : Unit identificator for each row.
  - `time` : Discrete observation times, starting at 1 for a given unit, until the end of its follow-up. The number of rows associated to a given unit corresponds to the follow-up duration.

  - `event` : Event indicator (1 if it occured, 0 otherwise) for given unit at a given time.
  - `z1, z2, z3, z4, x1, x2, x3, x4` : Covariate values for a given unit at a given time.
- `data.summary` : Data frame with n rows containing summarized information on the generated data for each unit:
  - `id` : Unit identificator (the ith row corresponding to the ith unit).
  - `t.obs` : Observed event or right-censoring time.
  - `delta` : Event indicator (1 if it occured, 0 otherwise).
  - `t.true` : True (possibly unobserved) event time (Inf for a cured unit).
  - `t.cens` : True (possibly unobserved) right-censoring time.
  - `cured` : True (possibly unobserved) cure status.
- `parameters` : List containing the defining elements of the tvcure model:
  - `beta` : The regression parameters in the long-term survival (or quantum) submodel.
  - `gam` : The regression parameters in the short-term survival (or timing) submodel.
  - `beta.f` : The multiplying coefficients of the additive terms in the long-term survival (or quantum) submodel.
  - `gam.f` : The multiplying coefficients of the additive terms in the short-term survival (or timing) submodel.
  - `f.theta` : A list of length 2 containing the functions defining the additive terms in the long-term survival (or quantum) submodel.
  - `f.gam` : A list of length 2 containing the functions defining the additive terms in the short-term survival (or timing) submodel.
  - `f0` : Density function governing the dynamic of the reference cumulative hazard on (0,Tmax).
  - `F0` : CDF governing the dynamic of the reference cumulative hazard on (0,Tmax).
- `call` : Function call.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Regression parameters
beta = c(beta0=.4, beta1=-.2, beta2=.15) ##  beta0 tunes the cure rate
gam = c(gam1=.2, gam2=.2)
beta.f = gam.f = rep(1,2) ## Multiply additive terms by 1.0 (by default)
## Data simulation
temp = simulateTVcureData2(n=500, seed=123,
                           beta=beta, gam=gam,
```

```
                          beta.f=beta.f, gam.f=gam.f,
                          RC.dist="exponential",mu.cens=550)
head(temp$rawdata) ## Overview of the simulated raw data
head(temp$data.summary) ## Overview of the summarized data
with(temp, c(cure.rate=cure.rate,RC.rate=RC.rate)) ## Cure and RC rates
```

`tvcure` *Fit of a tvcure model.*

Description

Fit of a double additive cure survival model with exogenous time-varying covariates.

Usage

```
tvcure(formula1, formula2, data,
      model0=NULL, noestimation=FALSE,
      baseline=c("S0","F0"), K0=20, pen.order0=2,
      K1=10, pen.order1=2, K2=10, pen.order2=2,
      phi.0=NULL, beta.0=NULL, gamma.0=NULL,
      a.tau=1, b.tau=1e-6, a.pen=1, b.pen=1e-4,
    tau.0=NULL, tau.min=1, tau.method = c("LPS","LPS2","Schall","grid","none"),
      psi.method = c("LM","NR","none"),
      lambda1.0=NULL, lambda1.min=1, lambda2.0=NULL, lambda2.min=1,
      lambda.method=c("LPS","LPS2","LPS3","nlminb","none"),
      logscale=FALSE,
      observed.hessian=TRUE, use.Rfast=TRUE, Wood.test=FALSE,
      ci.level=.95,
      criterion=c("logEvid","deviance","lpen","AIC","BIC","gradient"),
      criterion.tol=1e-2, grad.tol=1e-2,
      RDM.tol=1e-4, fun.tol=1e-3, Lnorm=c("Linf","L2"),
      iterlim=50, iter.verbose=FALSE, verbose=FALSE)
```

Arguments

`formula1`	A formula describing the linear predictor in the long-term (cure) survival (or quantum) submodel.
`formula2`	A formula describing the linear predictor in the short-term (cure) survival (or timing) submodel.
`data`	A data frame with survival data in a counting process format. It should always contain at least the following entries: • `id` : the <id> of the unit associated to the data in a given line in the data frame.

- `time` : the integer time at which the observations are reported. For a given unit, it should be a sequence of CONSECUTIVE integers starting at 1 for the first observation.
- `event` : a sequence of 0-1 event indicators. For the lines corresponding to a given unit, it starts with 0 values concluded by a 0 in case of right-censoring or by a 1 if the event is observed at the end of the follow-up.

`model0`	(Optional) tvcure object from which starting values for the regression parameters, spline and penalty parameters are extracted. Make sure that it corresponds to the same model specification. The values of its components are overhidden by `phi.0`, `beta.0`, `gamma.0`, `tau.0`, `lambda1.0`, `lambda2.0` when they are not NULL. That possibility can be found useful to accelerate the fit of the same model on other data or for specific changes in model parameters (such as penalty parameters). (Default: NULL).
`noestimation`	Logical specifying that regression, spline and penalty parameters should not be estimated, but rather be fixed at their initial values (as for example provided by `model0` or other entries). (Default: FALSE).
`baseline`	Baseline ("S0" or "F0") used to specify the dependence of the cumulative hazard dynamics on covariates (Default: "S0"): Baseline S0: $S(t\|x) = S_0(t)^{\exp^{\gamma' x}}$; Baseline F0: $F(t\|x) = F_0(t)^{\exp \gamma' x}$
`K0`	Number of B-splines used to specify $\log f_0(t)$ (Default: 20).
`pen.order0`	Penalty order for the P-splines used to specify $\log f_0(t)$ (Default: 2).
`K1`	Number of P-splines for a given additive term in the long-term (or quantum) survival sumbodel (Default: 10).
`pen.order1`	Penalty order for the P-splines in the long-term survival (or quantum) sumbodel (Default: 2).
`K2`	Number of P-splines for a given additive term in the short-term (or timing) survival sumbodel (Default: 10).
`pen.order2`	Penalty order for the P-splines in the short-term survival (or timing) sumbodel (Default: 2).
`phi.0`	(Optional) vector of length `K0` with starting values for the P-spline parameters in $\log f_0(t)$.
`beta.0`	(Optional) starting value for the regression and spline parameters in the long-term survival (or quantum) submodel.
`gamma.0`	(Optional) starting value for the regression and spline parameters in the short-term survival (or timing) submodel.
`a.tau`	Hyperprior parameter in the Gamma(a.tau,b.tau) prior for the penalty parameter τ tuning the smoothness of $\log f_0(t)$ (Default: 1.0).
`b.tau`	Hyperprior parameter in the Gamma(a.tau,b.tau) prior for the penalty parameter τ tuning the smoothness of $\log f_0(t)$ (Default: 1e-6).
`a.pen`	Hyperprior parameter in the Gamma(a.pen,b.pen) priors for the penalty parameters λ_1 and λ_2 tuning the smoothness of the additive terms in the long-term (quantum) and short-term (timing) survival submodels. (Default: 1.0).

`b.pen`	Hyperprior parameter in the Gamma(a.pen,b.pen) priors for the penalty parameters λ_1 and λ_2 tuning the smoothness of the additive terms in the long-term (quantum) and short-term (timing) survival submodels. (Default: 1e-4).
`tau.0`	Starting value for τ.
`tau.min`	Minimal value for the penalty parameter τ. (Default: 1.0).
`tau.method`	Method used to calculate the posterior mode of $p(\tau\|\mathcal{D})$: "LPS", "LPS2", "Schall" (Fellner-Schall algorithm), "grid" (best choice in a regular grid on the log-scale) or "none" (stick to the initial value tau.0). LPS and LPS2, based on Laplace P-splines, both maximize the marginal posterior of the penalty parameter τ using a fixed-point method, with LPS relying on the prior calculation of eigenvalues. (Default: "LPS").
`psi.method`	Algorithm used for the computation of the conditional posterior mode of the regression and splines parameters. Possible choices are Levenberg-Marquardt ("LM"), Newton-Raphson ("NR") or "none" (when the coefficients remain fixed at their initial values).
`lambda1.0`	(Optional) J1-vector with starting values for the penalty parameters of the additive terms in the long-term survival (or quantum) submodel.
`lambda1.min`	Minimal value for the J1 penalty parameters λ_1 of the additive terms in the long-term survival (or quantum) submodel. (Default: 1.0).
`lambda2.0`	(Optional) J2-vector with starting values for the penalty parameters of the additive terms in the short-term survival (or timing) submodel.
`lambda2.min`	Minimal value for the J2 penalty parameters λ_2 of the additive terms in the short-term survival (or timing) submodel. (Default: 1.0).
`lambda.method`	Method used ("LPS", "LPS2", "LPS3", "nlminb" or "none") to select the penalty parameters of the additive terms in the long-term survival (or quantum) submodel: • `LPS`, `LPS2`, or `LPS3` : based on Laplace P-splines where the marginal posterior of the penalty parameters is maximized using a fixed-point method. LPS is based on the prior calculation of eigenvalues (unlike LPS2) and delivers results of comparable quality to those of nlminb, but much more quickly. LPS3 work sequentially and separately on long- and short-term parameters with potentially convergence issues ; • `Schall` : Fellner-Schall method ; • `nlminb` : nonlinear maximization of the marginal posterior of the penalty parameters using the R function <nlminb> ; • `none` : penalty parameters are set at their initial values.
`logscale`	Logical: when TRUE, select λ_1 or λ_2 by maximizing $p(\log(\lambda_k)\|D)$, maximize $p(\lambda_k\|D)$ otherwise. (Default= FALSE).
`observed.hessian`	Logical indicating if a fast approximation of the Hessian matrix based on cross-products is preferred over its expected value. (Default: TRUE).
`use.Rfast`	Logical indicating if matrix functions from the Rfast package should be used to fasten computation. (Default: TRUE).

`Wood.test`	Logical indicating if P-values based on Wood's test (Biometrika 2013) of the significance of additive terms should be preferred over basic Chi-square tests. (Default: FALSE).
`ci.level`	Default value for the levels of the credible intervals. (Default: 0.95).
`criterion`	Criterion used to assess convergence of the estimation procedure (Default: "logEvid"): • `logEvid` : log of the evidence (also named *marginal likelihood* or *model likelihood*), i.e. the log of the marginal posterior of the penalty parameters at their selected values ; • `deviance` : deviance or -2 log(Likelihood) ; • `AIC` : Akaike information criterion ; • `BIC` : Bayesian (or Schwarz) information criterion ; • `gradient` : Lp-norm of the gradient of the log of the joint posterior w.r.t. the regression and spline parameters.
`criterion.tol`	Maximum absolute difference between the successive values of the `criterion` values (when different from "gradient") to declare convergence. (Default: 1e-2).
`grad.tol`	Tolerance threshold for the absolute value of each gradient component when monitoring convergence. (default: 1e-2).
`RDM.tol`	Tolerance thershold for the Relative Damping Measure (= RDM) when monitoring convergence (default: 1e-4).
`fun.tol`	Tolerance threshold for variations in the maximized function during the final iterations of posterior mode computation and convergence monitoring (default: 1e-3).
`Lnorm`	Lp norm used to evaluate the gradient for convergence assessment. Options are "Linf" (default) or "L2".
`iterlim`	Maximum number of iterations. (Default: 50).
`iter.verbose`	Logical indicating if the values of the convergence criterions should be printed after each iteration. (Default: FALSE).
`verbose`	Logical indicating if additional output based on gradients should be printed at the end of each iteration. (Default: FALSE).

Value

An object of type `tvcure.object`.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

Examples

```
require(tvcure)
## Simulated data generation
beta = c(beta0=.4, beta1=-.2, beta2=.15) ; gam = c(gam1=.2, gam2=.2)
data = simulateTVcureData(n=500, seed=123, beta=beta, gam=gam,
                          RC.dist="exponential",mu.cens=550)$rawdata
## TVcure model fitting
tau.0 = 2.7 ; lambda1.0 = c(40,15) ; lambda2.0 = c(25,70) ## Optional
model = tvcure(~z1+z2+s(x1)+s(x2), ~z3+z4+s(x3)+s(x4), data=data,
               tau.0=tau.0, lambda1.0=lambda1.0, lambda2.0=lambda2.0)
print(model)
plot(model, pages=1)
```

`tvcure.object` *Object resulting from the fit of a tvcure model using function 'tvcure'.*

Description

An object returned by the `tvcure` function: this is a list with various components related to the fit of such a model.

Value

A `tvcure_object` is a list with following elements:

- `formula1` : A formula describing the linear predictor in the long-term (cure) survival (or quantum) submodel.
- `formula2` : A formula describing the linear predictor in the short-term (cure) survival (or timing) submodel.
- `baseline` : Baseline ("S0" or "F0") used to specify the dependence of the cumulative hazard dynamics on covariates.
- `id` : the <id> of the unit associated to the data in a given line in the data frame.
- `time` : the integer time at which the observations are reported. For a given unit, it should be a sequence of CONSECUTIVE integers starting at 1 for the first observation.
- `event` : a sequence of 0-1 event indicators. For the lines corresponding to a given unit, it starts with 0 values concluded by a 0 in case of right-censoring or by a 1 if the event is observed at the end of the follow-up.
- `regr1` : List returned by `DesignFormula` when evaluated on `formula1`.
- `regr2` : List returned by `DesignFormula` when evaluated on `formula2`.
- `K0` : Number of B-splines used to specify $\log f_0(t)$.
- `fit` : A list containing different elements describing the fitted tvcure model:
  - `llik` : Log likelihood value of the fitted tvcure model at convergence.
  - `lpen` : Log of the penalized joint posterior at convergence.

  - `dev` : Deviance of the fitted tvcure model at convergence.
  - `mu.ij` : Expected value $\mu_{ij} = h_p(t_{ij}|z(t_{ij}), x(t_{ij}))$ for the event indicator of unit i at time t_{ij}.
  - `res` : Standardized residual $(d_{ij}-\mu_{ij})/\sqrt{\mu_{ij}}$ for unit i at time t_{ij} where $\mu_{ij} = h_p(t_{ij}|z(t_{ij}), x(t_{ij}))$ and d_{ij} is the event indicator.
  - `phi` : Vector of length K_0 containing the estimated B-splines coefficients in $\log f_0(t)$.
  - `marginalized` : Marginalization indicator (over penalty parameters) when reporting regression and spline parameter estimates.
  - `nbeta` : Number of regression and spline parameters in the long-term (cure) survival (or quantum) submodel.
  - `ci.level` : Selected level for credible intervals.
  - `beta` : (nbeta x 6) matrix containing the point estimates, standard errors, credible intervals, Z-scores and P-values of the regression and spline parameters in the long-term (cure) survival (or quantum) submodel.
  - `ngamma` : Number of regression and spline parameters in the short-term (cure) survival (or timing) submodel.
  - `gamma` : (ngamma x 6) matrix containing the point estimates, standard errors, credible intervals, Z-scores and P-values of the regression and spline parameters in the short-term (cure) survival (or timing) submodel.
  - `gam` : ngamma-vector with the point estimates of the regression and spline parameters in the short-term (cure) survival (or timing) submodel.
  - `grad.beta` : Gradient of the log joint posterior of <beta>, the regression and spline parameters in the long-term (cure) survival (or quantum) submodel.
  - `Hes.beta` : Hessian of the log joint posterior of <beta>.
  - `Hes.beta0` : Hessian of the log joint posterior of <beta> (with the roughness penalty part omitted).
  - `grad.gamma` : Gradient of the log joint posterior of <gamma>, the regression and spline parameters in the short-term (cure) survival (or timing) submodel.
  - `Hes.gamma` : Hessian of the log joint posterior of <gamma>.
  - `Hes.gamma0` : Hessian of the log joint posterior of <gamma> (with the roughness penalty part omitted).
  - `Mcal.1` : Hessian of the log joint posterior of the spline parameters in <beta> conditionally on the non-penalized parameters.
  - `Mcal.2` : Hessian of the log joint posterior of the spline parameters in <gamma> conditionally on the non-penalized parameters.
  - `Hes.betgam` : (nbeta x ngamma) matrix with the cross derivatives of the log joint posterior of (<beta>,<gamma>).
  - `grad.regr` : Gradient of the log joint posterior of <beta,gamma>.
  - `Hes.regr` : Hessian of the log joint posterior of <beta,gamma>.
  - `Hes.regr0` : Hessian of the log joint posterior of <beta,gamma> (with the roughness penalty part omitted).
  - `grad.phi` : Gradient of the log joint posterior of <phi>, the spline parameters in $\log f_0(t)$.
  - `Hes.phi` : Hessian of the log joint posterior of <phi>.
  - `Hes.phi0` : Hessian of the log joint posterior of <phi> (with the roughness penalty part omitted).

- `T` : Follow-up time after which a unit is declared cured in the absence of a past event.
- `t.grid` : Grid of discrete time values on (1,T): 1,...,T.
- `f0.grid` : Estimated values for $f_0(t)$ on `t.grid`.
- `F0.grid` : Estimated values for $F_0(t)$ on `t.grid`.
- `S0.grid` : Estimated values for $S_0(t)$ on `t.grid`.
- `dlf0.grid` : (ngrid x length(phi)) matrix with the jth line containing the gradient of $\log f_0(t_j)$ w.r.t. <phi>.
- `dlF0.grid` : (ngrid x length(phi)) matrix with the jth line containing the gradient of $\log F_0(t_j)$ w.r.t. <phi>.
- `dlS0.grid` : (ngrid x length(phi)) matrix with the jth line containing the gradient of $\log S_0(t_j)$ w.r.t. <phi>.
- `k.ref` : Index of the reference component in <phi> set to 0.0.
- `a, b` : Hyperparameters of the Gamma(a,b) prior for the penalty parameters of the additive terms.
- `criterion` : Criterion used to assess convergence of the estimation procedure.
- `grad.psi` : Gradient of the log joint posterior of <phi[-k.ref]>, i.e. the spline parameters in $\log f_0(t)$ with the fixed reference component omitted.
- `Hes.psi0` : Hessian of the log joint posterior of <phi[-k.ref]> (with the roughness penalty part omitted).
- `Hes.psi` : Hessian of the log joint posterior of <phi[-k.ref]>.
- `tau` : Selected value for the penalty parameter τ tuning the smoothness of $\log f_0(t)$.
- `pen.order0` : Penalty order for the P-splines used to specify $\log f_0(t)$.
- `logscale` : Logical: when TRUE, select λ_1 or λ_2 by maximizing $p(\log(\lambda_k)|D)$, maximize $p(\lambda_k|D)$ otherwise. (Default= TRUE).
- `lambda1` : Selected values for the penalty parameters λ_1 tuning the smoothness of the additive terms in the long-term (cure) survival (or quantum) submodel.
- `pen.order1` : Penalty order for the P-splines in the long-term survival (or quantum) submodel.
- `lambda2` : Selected values for the penalty parameters λ_2 tuning the smoothness of the additive terms in the short-term (cure) survival (or timing) submodel.
- `pen.order2` : Penalty order for the P-splines in the short-term survival (or timing) submodel.
- `tau.method` : Method used to calculate the posterior mode of $p(\tau_0|\mathcal{D})$.
- `lambda.method` : Method used to select the penalty parameters of the additive terms in the long-term survival (or quantum) submodel.
- `ED1` : Effective degrees of freedom for each of the additive terms in the long-term survival (or quantum) submodel, with the selected statistical test for significance and its P-value.
- `ED2` : Effective degrees of freedom for each of the additive terms in the short-term survival (or timing) submodel, with the selected statistical test for significance and its P-value.
- `ED1.Tr` : Effective degrees of freedom for each of the additive terms in the long-term survival (or quantum) submodel, with Wood's statistical test for significance and its P-value.
- `ED2.Tr` : Effective degrees of freedom for each of the additive terms in the short-term survival (or timing) submodel, with Wood's statistical test for significance and its P-value.

  - `ED1.Chi2` : Effective degrees of freedom for each of the additive terms in the long-term survival (or quantum) submodel, with a Chi-square test for significance and its P-value.
  - `ED2.Chi2` : Effective degrees of freedom for each of the additive terms in the short-term survival (or timing) submodel, with a Chi-square test for significance and its P-value.
  - `nobs` : Total number of observations.
  - `n` : Total number of units or subjects.
  - `d` : Total number of observed events.
  - `ED1.tot` : Total effective degrees of freedom for the long-term survival (or quantum) submodel.
  - `ED2.tot` : Total effective degrees of freedom for the short-term survival (or timing) submodel.
  - `ED.tot` : Total effective degrees of freedom for the tvcure model.
  - `AIC` : Akaike information criterion for the fitted model with a penalty calculated using the total effective degrees of freedom, -2log(L) + 2*ED.tot, larger values being preferred during model selection.
  - `BIC` : Bayesian (Schwarz) information criterion for the fitted model with a penalty calculated using the total effective degrees of freedom and the total number of observed events, -2log(L) + log(d)*ED.tot, smaller values being preferred during model selection.
  - `levidence` : Log-evidence of the fitted model, larger values being preferred during model selection.
  - `iter` : Number of iterations required to achieve convergence.
  - `elapsed.time` : Total duration (in seconds) of the estimation procedure.
- `call` : Function call.
- `converged` : Binary convergence status.
- `logLik` : Log-likelihood of the fitted model.

Author(s)

Philippe Lambert <`p.lambert@uliege.be`>

References

Lambert, P. and Kreyenfeld, M. (2025). Time-varying exogenous covariates with frequently changing values in double additive cure survival model: an application to fertility. *Journal of the Royal Statistical Society, Series A*. <doi:10.1093/jrsssa/qnaf035>

See Also

`tvcure`, `print.tvcure`, `plot.tvcure`

Index