Package 'TBSSurvival'

January 5, 2017

Version 1.3

Date 2017-01-05

Title Survival Analysis using a Transform-Both-Sides Model

Author Adriano Polpo <polpo@ufscar.br>, Cassio de Campos <cassiopc@acm.org>, D. Sinha <sinhad@stat.fsu.edu>, Stuart Lipsitz <slipsitz@partners.org>, Jianchang Lin <jlin@stat.fsu.edu>

Maintainer Cassio de Campos <cassiopc@acm.org>

Depends R (>= 3.3.0)

Imports normalp, survival, mcmc, R.utils, coda, Rsolnp, BMS

Description Functions to perform the reliability/survival analysis using a parametric Transform-both-sides (TBS) model.

License GPL (>= 3)

Repository CRAN

URL http://code.google.com/p/tbssurvival/

NeedsCompilation no

Date/Publication 2017-01-05 14:45:36

R topics documented:

`alloyT7987` *Data set Alloy T7987*

Description

Alloy - T7987: data extracted from Meeker and Escobar (1998), pp. 131.

Value

The two data variables are:

`time`	Time of failure of the specimen.
`delta`	censoring indication (0 means right-censored, 1 means no censoring).

References

Meeker, W. and Escobar, L. (1998) *Statistical Methods for Reliability Data*. Willey, ISBN 0-471-14328-6.

See Also

`tbs.survreg.mle`

Examples

```
## See \link{tbs.survreg.mle} and \link{tbs.survreg.be}.

## The data set looks like this:
## time delta
##  94     1
##  96     1
##  99     1
##  99     1
## 104     1
## 108     1
## 112     1
## 114     1
## 117     1
## 117     1
```

dist.error *The Wrapper for error distribution functions to use with TBS*

Description

Constructor of a list of density function, distribution function, quantile function, random generation and hazard function for the Transform-Both-Sides (TBS) distribution.

Usage

```
dist.error(dist="norm")
```

Arguments

dist	Distribution of error, dist = "norm", "t", "doubexp", "cauchy" or "logistic". A choice "all" can also be given, in which case a list with all the available distributions will be generated.

Details

This wrapper creates a list with the necessary information about a distribution to be used with the Transform-Both-Sides (TBS) model. It returns a list with five items, namely the density function, distribution function, quantile function, random generation function, and name (a string). For example, with dist = "norm" it gives list(dnorm,pnorm,qnorm,rnorm,"norm"). The idea is that the user can implement its own list to be used with the TBS, as long as it is a zero-centered unimodal symmetric distribution, and replace the call of dist.error with their own list (given in a similar way as the output of dist.error).

Value

A list(d.dist,p.dist,q.dist,r.dist,name.dist) according with the chosen distribution. The functions d.dist,p.dist,q.dist,r.dist have to accept exactly two arguments: the first is the actual argument to the function, and the second is a parameter.

Examples

```
## this will return list(dcauchy,pcauchy,qcauchy,rcauchy,"cauchy")
dist = dist.error("cauchy")

## a user-built distribution would look like:
## dist = list(
##         d = function(x,xi) dmydistrib(x,param=xi), # density
##         p = function(x,xi) pmydistrib(x,param=xi), # distr
##         q = function(x,xi) qmydistrib(x,param=xi), # quantile
##         r = function(x,xi) rmydistrib(x,param=xi), # generation
##         name = "mydistrib"
##        )
```

tbs *The TBS Time Failure Distribution*

Description

Density function, distribution function, quantile function, random generation function and hazard function for the Transform-Both-Sides (TBS) model.

Usage

```
dtbs(time,lambda=1,xi=1,beta=1,x=NULL,dist=dist.error("norm"))
ptbs(time,lambda=1,xi=1,beta=1,x=NULL,dist=dist.error("norm"))
qtbs(p,lambda=1,xi=1,beta=1,x=NULL,dist=dist.error("norm"))
rtbs(n,lambda=1,xi=1,beta=1,x=NULL,dist=dist.error("norm"))
htbs(time,lambda=1,xi=1,beta=1,x=NULL,dist=dist.error("norm"))
```

Arguments

time	vector of quantiles.
p	vector of probabilities.
n	number of observations.
lambda	parameter of TBS.
xi	parameter of the error distribution.
beta	parameter of the linear regressor.
x	vector/matrix of co-variables, x=NULL if there are not co-variables.
dist	Distribution of error, it can be string such as dist = "norm", "t", "doubexp", "cauchy" or "logistic, or it can also be given as a list of functions (density, distribution, quantile, random generation, name). Details below.

Details

The density function, distribution function, quantile function, random generation and hazard function for the failure time of a TBS Model. The distribution of error can be chosen from Normal, t-Student, Cauchy, Logistic and Doub-Exponential (Laplace), or can be given by the user (as long as it is zero-centered, unimodal and symmetric – TBS does not check it). See the help of dist.error for examples.

Value

'dtbs' gives the density, 'ptbs' gives the distribution function, 'qtbs' gives the quantile function, 'rtbs' generates random deviates, 'htbs' gives the hazard function.

See Also

dist.error

Examples

```
ptbs(1,lambda=2,xi=1,beta=1,dist=dist.error("norm"))
```

`tbs.survreg.be` *Bayesian Estimation of the TBS Model for Survival Data*

Description

This function performs the Bayesian estimation of the Transform-Both-Sides (TBS) model. The priors for the parameters 'lambda' and 'xi' are uniform-exponential mixtures and, if not specified, for parameter beta is a normal with mean 5 and sd 5. The estimations are done by Metropolis-Hasting (using the function 'metrop' availible with the package 'mcmc').

Usage

```
tbs.survreg.be(formula, dist=dist.error("norm"),max.time = -1, guess.beta = NULL,
              guess.lambda = 1, guess.xi = 1, burn = 1000, jump = 2, size = 500,
                scale = 0.1, prior.mean = NULL, prior.sd = NULL, seed = 1234)
```

Arguments

`formula`	A formula specification containing a Surv model with right-censored (or no censored) data as in the package survival.
`dist`	Error distribution; `dist` can be given by name (`"norm"`, `"doubexp"`, `"t"`, `"cauchy"` or `"logistic"`) or by dist.error.
`max.time`	Maximum time (in minutes) to run the optimization (<= 0 means no limit).
`guess.beta`	Initial value of the Markov Chain for the vector 'beta'. Default will fill it with zeros.
`guess.lambda`	Initial value of the Markov Chain for the parameter 'lambda'.
`guess.xi`	Initial value of the Markov Chain for the parameter 'xi'.
`burn`	Burn-in: number of initial samples of the posterior not to use.
`jump`	Number of jumps between each sample of the posterior to avoid the problem of auto-correlation between the samples.
`size`	Size of final sample of the posterior.
`scale`	Parameter of 'metrop' function. Controls the acceptance rate.
`prior.mean`	Prior Mean for the MCMC.
`prior.sd`	Prior std deviation for the MCMC.
`seed`	The number that is used to initialize the seed for random number generation.

Details

This function performs the Bayesian estimation of the Transform-Both-Sides (TBS) model. The priors for the parameters 'lambda' and 'xi' are uniform-exponential mixtures and, if not specified, for parameter beta is a normal with mean 5 and sd 5. The estimations are done by Metropolis-Hasting (using the function 'metrop' availible with the package 'mcmc').

Value

An element of the class tbs.survreg.be, with the components:

`call`	function evaluated.
`x`	co-variable matrix used.
`time`	survival time.
`delta`	censor status.
`post`	posterior sample of the parameters.
`lambda`	posterior mean of lambda.
`xi`	posterior mean of xi.
`beta`	vector with posterior mean of beta.
`lamda.sd`	standard deviation for lambda.
`xi.sd`	standard deviation of for xi.
`beta.sd`	standard deviation of for beta.
`lambda.HPD`	95% high posterior density credal interval of lambda.
`xi.HPD`	95% high posterior density credal interval of xi.
`beta.HPD`	95% high posterior density credal interval vector of beta.
`DIC`	Deviance Information Criterion.
`error`	summary statistics for the posterior of error of TBS model.
`error.dist`	error distribution.
`run.time`	Time spent with the function.

References

Meeker, W. and Escobar, L. (1998) *Statistical Methods for Reliability Data*. Willey, ISBN 0-471-14328-6.

See Also

`dist.error`, `tbs.survreg.mle`, `dtbs`, `ptbs`, `qtbs`, `rtbs`.

Examples

```
# set.seed is used to produce the same results all times.
set.seed(1234)

# Alloy - T7987: data extracted from Meeker and Escobar (1998), pp. 131)
data(alloyT7987)
alloyT7987$time  <- as.double(alloyT7987$time)
alloyT7987$delta <- as.double(alloyT7987$delta)

# Bayesian estimation with logistic error
formula <- survival::Surv(alloyT7987$time,alloyT7987$delta == 1) ~ 1
tbs.be <- tbs.survreg.be(formula,guess.lambda=1,guess.xi=1,guess.beta=5,
                     dist=dist.error("logistic"),burn=1000,jump=10,size=500,scale=0.06)
```

```
# Kapan-Meier estimator
km <- survival::survfit(formula = survival::Surv(alloyT7987$time, alloyT7987$delta == 1) ~ 1)

# Plot survival function
plot(tbs.be,lwd=2,HPD=TRUE,HPD.alpha=0.95,col.HPD=2,lty.HPD=1,lwd.HPD=2)
lines(km)

# Plot survival function
plot(tbs.be,plot.type="hazard",lwd=2,HPD=TRUE,HPD.alpha=0.95,col.HPD=2,lty.HPD=1,lwd.HPD=2)

# Plot auto-correlation of the posterior sample
plot(tbs.be,plot.type="auto")

# Plot "time-series" of the posterior sample
plot(tbs.be,plot.type="ts")
```

`tbs.survreg.mle` *MLE of the TBS Model for Failure Data*

Description

This function performs the Maximum Likelihhod Estimation of the TBS model. The optimization is done by the function 'optim' (or optionally the package Rsolnp when available).

Usage

```
tbs.survreg.mle(formula,dist=dist.error("all"),
                method=c("Nelder-Mead", "BFGS", "Rsolnp", "SANN", "CG"),
                verbose=FALSE,nstart=10,max.time=-1,seed=1234,gradient=FALSE)
```

Arguments

`formula`	A formula specification containing a Surv model with right-censored data as in the package survival.
`dist`	error distribution; `dist` can be given by name (`"norm"`, `"doubexp"`, `"t"`, `"cauchy"` or `"logistic"`) or by dist.error.
`method`	a vector of numerical methods to be used in the optimization. The function try all listed methods and returns all results, together with an indication of the solution with maximal likelihood among them.
`verbose`	Boolean to indicate the amount of output during the execution of the optimization.
`nstart`	Number of feasible initial points to guess when performing the optimization.
`max.time`	Maximum time (in minutes) to run the optimization (<= 0 means no limit).
`seed`	The number that is used to initialize the seed for random number generation.
`gradient`	if TRUE, MLE tries to use the implemented gradient functions (usually the numerical ones are ok).

Details

This function calls numerical optimization methods to maximize the likelihood of the TBS model, according to the given error distribution, method of optimization, and formula. The formula is supposed to have a Surv object and possibility co-variates, just as the standard specification of R formulas. The optimizers are going to do their best to find high likelihood estimates, but as in most estimation methods that need a numerical optimization procedure, the obtained estimate cannot be guaranteed to be a global optimal solution, but instead is dependent on the initial guessing points, and thus on the seed of the random number generation.

Value

Either an element of class tbs.survreg.mle (with print, summary, and plot functions) or a list of them (depending whether the call of tbs.survreg.mle was made for a single distribution or a list of them). In case it is a list, additional fields named best and best.n give the name and the position of the best estimation in the list, respectively. Each element of tbs.survreg.mle has the following components:

`lambda`	The estimate for parameter lambda
`xi`	The estimate for parameter xi
`beta`	A vector with the estimate for parameter beta
`lambda.se`	The standard error for parameter lambda
`xi.se`	The standard error for parameter xi
`beta.se`	A vector with the standard error for parameter beta
`log.lik`	The log-likelihood at parameters `par`.
`error.dist`	The error distribution chosen.
`AIC`	Akaike Information Criterion.
`AICc`	AICc is AIC with a second order correction for small sample sizes.
`BIC`	Bayesian Information Criterion.
`method`	Numerical method used to achive the MLE.
`convergence`	If `convergence` is `FALSE` then it was not possible to find the MLE.
`time`	observed survival times.
`error`	error of the estimated model.
`call`	function evaluated.
`formula`	formula entered by user.
`run.time`	Time spent with the function.

References

Meeker, W. and Escobar, L. (1998) *Statistical Methods for Reliability Data*. Willey, ISBN 0-471-14328-6.

See Also

`dist.error`, `tbs.survreg.be`, `dtbs`, `ptbs`, `qtbs`, `rtbs`.

Examples

```
# Alloy - T7987: data extracted from Meeker and Escobar (1998), pp. 131.
data(alloyT7987)
alloyT7987$time <- as.double(alloyT7987$time)
alloyT7987$delta <- as.double(alloyT7987$delta)

# MLE estimation with logistic error
formula <- survival::Surv(alloyT7987$time,alloyT7987$delta == 1) ~ 1
tbs.mle <- tbs.survreg.mle(formula,dist=dist.error("logistic"),method="Nelder-Mead",nstart=3)

# Kaplan-Meier estimation
km <- survival::survfit(formula)

# Plot survival function
plot(tbs.mle)
lines(km)

# Plot hazard function
plot(tbs.mle, plot.type="hazard")
```

Index