Package ‘TBSSurvival’

July 1, 2012

Version 1.0

Date 2012-06-30

Title TBS Model R package

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

Maintainer Cassio de Campos <cassio@idsia.ch>

Depends R (>= 2.14.0), stats, mcmc, coda, survival, normalp, R.utils

Suggests multicore, snowfall, Rsolnp

Description This package contais the functions used to perform the reliability analysis using a parametric Transform-both-sides (TBS) model.

Repository CRAN

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

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}.
```

tbs *The TBS Time Failure Distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the TBS distribution.

Usage

```
dtbs(time,lambda=1,xi=1,beta=1,x=NULL,dist="norm")
ptbs(time,lambda=1,xi=1,beta=1,x=NULL,dist="norm")
qtbs(p,lambda=1,xi=1,beta=1,x=NULL,dist="norm")
rtbs(n,lambda=1,xi=1,beta=1,x=NULL,dist="norm")
htbs(time,lambda=1,xi=1,beta=1,x=NULL,dist="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, dist = "norm", "t", "doubexp", "cauchy" or "logistic".

Details

The density, 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).

Value

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

Examples

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

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

Description

This function perform the Bayesian estimation of the TBS model. The prior for the parameters 'lambda' and 'xi' are uniform-exponential mixture and, if not specified, for parameter beta0 is a normal with mean 5 and sd 5. The estimations are done by metroplois-hasting (using the function 'metrop' availible at package 'mcmc').

Usage

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

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` = `"norm"`, `"doubexp"`, `"t"`, `"cauchy"` or `"logistic"`.
`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'.

`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 firsts samples of posterior to not use.
`jump`	number of jump between each sample of posterior to avoid the problem of auto-correlation between the samples.
`size`	size of final sample of 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.

Details

This function was developed to the problem without co-variables. Some changes maybe are need to work with co-variables.

Value

A list with components:

`call`	function evaluated.
`x`	co-variable matrix used.
`time`	survival time.
`delta`	censor status.
`post`	posterior sample of the parameters.
`par`	the parameters are estimated as median of posterior sample.
`par.sd`	standard deviation of posterior sample.
`par.HPD`	95% high posterior density credal interval of each parameter.
`DIC`	Deviance Information Criterion.
`error`	summary statistics and high posterior density credal interval of 95% for the posterior of error of TBS model.
`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

`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 <- Surv(alloyT7987$time,alloyT7987$delta == 1) ~ 1
tbs.be <- tbs.survreg.be(formula,guess.lambda=1,guess.xi=1,guess.beta=5,
                         dist="logistic",burn=1000,jump=10,size=1000,scale=0.06)
# evaluating the estimated survival and HPD
axis.x <- seq(0.01,300,1)
aux <- matrix(NA,length(axis.x),length(tbs.be$post[,1]))
for (j in 1:length(tbs.be$post[,1])) {
  aux[,j] <- 1-ptbs(axis.x,lambda=tbs.be$post[j,1],xi=tbs.be$post[j,2],
                    beta=tbs.be$post[j,3],dist="logistic")
}
survival <- matrix(NA,length(axis.x),3)
for (i in 1:length(axis.x)) {
  survival[i,] <- c(mean(aux[i,]),HPDinterval(as.mcmc(aux[i,]),0.95))
}
rm(aux,i,j)

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

plot(km,ylab="",xlab="", xlim=c(min(alloyT7987$time),max(alloyT7987$time)),
     conf.int=FALSE,axes=FALSE,lty=1,lwd=1)
t <- seq(min(alloyT7987$time),max(alloyT7987$time),
         (max(alloyT7987$time)-min(alloyT7987$time)-0.01)/1000)
title(ylab="R(t)", xlab="t: number of cycles (in thousands)",
      sub="Alloy - T7987 (cf. Meeker and Escobar (1998), pp. 131)",
      main="Reliability function (BE)",cex.lab=1.2)
axis(1,at=c(93,100,150,200,250,300),lwd=2,lwd.ticks=2,pos=0)
axis(2,lwd=2,lwd.ticks=2,pos=min(alloyT7987$time))
legend(170,0.95,c("Kaplan-Meier",
                  expression(textstyle(paste("TBS / ",sep="")) ~ epsilon
                  ~ textstyle(paste("~",sep="")) ~ Logistic),
                  "0.95 HPD Interval"),
       col=c(1,2,2),lty=c(1,1,2),cex=1.1,lwd=c(1,2,2),bg="white")
lines(axis.x,survival[,1],type="l",lwd=2,col=2,lty=1)
lines(axis.x,survival[,2],type="l",lwd=2,col=2,lty=2)
lines(axis.x,survival[,3],type="l",lwd=2,col=2,lty=2)
```

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

Description

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

Usage

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

Arguments

`formula`	A formula specification containing a Surv model with right-censored data as in the package survival.
`dist`	error distribution; `dist = "norm"`, `"doubexp"`, `"t"`, `"cauchy"` or `"logistic"`.
`method`	a vector of numerical methods to be used in the optimization. The function try all listed methods and returns 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).
`ncore`	If parallel computing is available, ncore tells the number of cores to use.

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 points, and thus on the seed of the random number generation.

Value

A list with components:

`par`	The best set of parameters found, the first value of par is the estimate of lambda, the second value is the estimate of xi and the others are the beta parameter.
`std.error`	Standar error for MLE. If it is not possible to evaluate return `NA`
`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.
`KS`	Komogorov-Simirnov Statistic. It is not evaluated with matrix `x` of co-variables.
`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

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 <- Surv(alloyT7987$time,alloyT7987$delta == 1) ~ 1
tbs.mle <- tbs.survreg.mle(formula,dist="logistic",method="Nelder-Mead",nstart=3)

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

plot(km,ylab="",xlab="",xlim=c(min(alloyT7987$time),max(alloyT7987$time)),
     conf.int=FALSE,axes=FALSE,lty=1,lwd=1)
t <- seq(min(alloyT7987$time),max(alloyT7987$time),
         (max(alloyT7987$time)-min(alloyT7987$time)-0.01)/1000)
title(ylab="R(t)",xlab="t: number of cycles (in thousands)",
      main="Reliability function (MLE)",
      sub="Alloy - T7987 (cf. Meeker and Escobar (1998), pp. 131)",
      cex.lab=1.2)
axis(1,at=c(93,100,150,200,250,300),lwd=1,lwd.ticks=1,pos=0)
axis(2,lwd=1,lwd.ticks=1,pos=min(alloyT7987$time))
legend(200,0.95,c("Kaplan-Meier",
                  expression(textstyle(paste("TBS / ",sep="")) ~ epsilon
                             ~ textstyle(paste("~",sep="")) ~ Logistic)),
       col=c(1,2),lty=c(1,1),cex=1.1,lwd=c(1,2),bg="white")
lines(t,1-ptbs(t,lambda=tbs.mle$par[1],xi=tbs.mle$par[2],
               beta=tbs.mle$par[3],dist=tbs.mle$error.dist),type="l",
               lwd=2,col=2,lty=1)
```

Index