Package ‘LongCART’

April 25, 2020

Type Package

Title Recursive Partitioning for Longitudinal Data and Right Censored Data Using Baseline Covariates

Version 2.0

Date 2020-04-24

Author Madan G Kundu

Maintainer Madan G Kundu <madan_g.kundu@yahoo.com>

Depends R (>= 3.4.0), nlme, rpart, survival, magic, survminer

Imports Formula

Description

Constructs tree for continuous longitudinal data and survival data using baseline covariates as partitioning variables according to the 'LongCART' and 'SurvCART' algorithm, respectively.

License GPL (>= 2)

URL https://www.r-project.org

BugReports https://github.com/madanstat/LongCART/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-24 23:10:02 UTC

R topics documented:

ACTG175 2
GBSG2 3
KMPlot.SurvCART 4
LongCART 5
plot.SurvCART 8
plotLongCART 9
StabCat 11
StabCat.surv 13
StabCont 14

StabCont.surv 16
SurvCART 18
text.SurvCART 20
textLongCART 22

Index **24**

`ACTG175` *Converted AIDS Clinical Trials Group Study 175 (source: speff2trial package)*

Description

ACTG 175 was a randomized clinical trial to compare monotherapy with zidovudine or didanosine with combination therapy with zidovudine and didanosine or zidovudine and zalcitabine in adults infected with the human immunodeficiency virus type I whose CD4 T cell counts were between 200 and 500 per cubic millimeter.

Usage

```
data(ACTG175)
```

Format

A data frame with 6417 observations from 2139 patients on the following 24 variables.

pidnum patient ID number

age age in years at baseline

wtkg weight in kg at baseline

hemo hemophilia (0=no, 1=yes)

homo homosexual activity (0=no, 1=yes)

drugs history of intravenous drug use (0=no, 1=yes)

karnof Karnofsky score (on a scale of 0-100)

oprior non-zidovudine antiretroviral therapy prior to initiation of study treatment (0=no, 1=yes)

z30 zidovudine use in the 30 days prior to treatment initiation (0=no, 1=yes)

zprior zidovudine use prior to treatment initiation (0=no, 1=yes)

preanti number of days of previously received antiretroviral therapy

race race (0=white, 1=non-white)

gender gender (0=female, 1=male)

str2 antiretroviral history (0=naive, 1=experienced)

strat antiretroviral history stratification (1:antiretroviral naive, 2:greater than 1 but less than 52 weeks of prior antiretroviral therapy, 3: greater than 52 weeks)

symptom symptomatic indicator (0=asymptomatic, 1=symptomatic)

treat treatment indicator (0=zidovudine only, 1=other therapies)

offtrt indicator of off-treatment before 96 weeks (0=no,1=yes)

r missing CD4 T cell count at 96 weeks (0=missing, 1=observed)

cens indicator of observing the event in days

days number of days until the first occurrence of: (i) a decline in CD4 T cell count of at least 50 (ii) an event indicating progression to AIDS, or (iii) death.

arms treatment arm (0=zidovudine, 1=zidovudine and didanosine, 2=zidovudine and zalcitabine, 3=didanosine)

time time in weeks

cd4 CD4 T cell count

References

Hammer, S.M., et al. (1996), A trial comparing nucleoside monotherapy with combination therapy in HIV-infected adults with CD4 cell counts from 200 to 500 per cubic millimeter. New England Journal of Medicine, 335:1081-1090.

GBSG2 *German Breast Cancer Study Group 2 (source: TH.data package)*

Description

A data frame containing the observations from the GBSG2 study.

Usage

```
data(GBSG2)
```

Format

A data frame with 686 observations on the following 10 variables.

`horTh` hormonal therapy, a factor with levels `no` `yes`

`age` age in years

`menostat` menopausal status, a factor with levels `Pre` `Post`

`tsize` tumor size (in mm)

`tgrade` an ordered factor with levels `I` < `II` < `III`

`pnodes` number of positive nodes

`progrec` progesterone receptor (in fmol).

`estrec` estrogen receptor (in fmol).

`time` recurrence free survival time (in days).

`cens` censoring indicator (0- censored, 1- event).

References

Schumacher M, Bastert G, Bojar H, Huebner K, Olschewski M, Sauerbrei W, Schmoor C, Beyerle C, Neumann RL, Rauschecker HF. Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group. Journal of Clinical Oncology. 1994 Oct;12(10):2086-93.

Examples

```
data(GBSG2)
```

KMPlot.SurvCART *KM plot for SurvCART object*

Description

Generates KM plot for sub-groups (i.e., terminal nodes) associated with survival tree generated by SurvCART()

Usage

```
KMPlot.SurvCART(x, scale.time = 1, type = 1, ...)
```

Arguments

x	a fitted object of class "SurvCART", containing a survival tree.
scale.time	Divides the time variable by the factor of scale.time. For example, if times are collected in days, then to plot time in years, specify 365.25.
type	1 for KM plot of survival probabilities, 2 for KM plot of censoring probabilities
...	arguments to be passed to or from other methods.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

text.SurvCART, plot.SurvCART, SurvCART, StabCat.surv, StabCont.surv

Examples

```
#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART() with time-to-event distribution: exponential, censoring distribution: None
out<- SurvCART(data=GBSG2, patid=”subjid”, censorvar=”cens”, timevar=”time”,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
        event.ind=1,  alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)

#Plot KM plot of survival probabilities for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)

#Plot KM plot of censoring probabilities for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)
```

LongCART *Longitudinal CART with continuous response via binary partitioning*

Description

Recursive partitioning for linear mixed effects model with continuous univariate response variables per LonCART algorithm based on baseline partitioning variables (Kundu and Harezlak, 2019).

Usage

```
LongCART(data, patid, fixed, gvars, tgvars, minsplit=40,
         minbucket=20, alpha=0.05, coef.digits=2, print.lme=FALSE)
```

Arguments

`data` name of the dataset. It must contain variable specified for `patid` (indicating subject id), all the variables specified in the formula and the baseline partitioning variables.

`patid` name of the subject id variable.

`fixed` a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by `+` operators, on the right. Model with `-1` to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only `1` right to the ~ operator.

`gvars` list of partitioning variables of interest. Value of these variables should not change over time. Regarding categorical variables, only numerically coded categorical variables should be specified. For nominal categorical variables or factors, please first create corresponding dummy variable(s) and then pass through `gvars`.

`tgvars` types (categorical or continuous) of partitioning variables specified in `gvar`. For each of continuous partitioning variables, specify `0` and for each of the categorical partitioning variables, specify `1`. Length of `tgvars` should match to the length of `gvars`

`minsplit` the minimum number of observations that must exist in a node in order for a split to be attempted.

`minbucket` he minimum number of observations in any terminal node.

`alpha` alpha (i.e., nominal type I error) level for parameter instability test

`coef.digits` decimal points for displaying coefficients in the tree structure.

`print.lme` if `TRUE`, then summary of fitte model from `lme()` will be printed for each node.

Details

Construct regression tree based on heterogeneity in linear mixed effects models of following type: `Y_i(t)= W_i(t)theta + b_i + epsilon_{it}` where `W_i(t)` is the design matrix, `theta` is the parameter associated with `W_i(t)` and `b_i` is the random intercept. Also, `epsilon_{it} ~ N(0,sigma ^2)` and `b_i ~ N(0,sigma_u^2)`.

Value

`Treeout` contains summary information of tree fitting for each terminal nodes and non-terminal nodes. Columns of `Treeout` include "ID", the (unique) node numbers that follow a binary ordering indexed by node depth, `n`, the number of observations reaching the node, `yval`, the fitted model of the response at the node, var, a factor giving the names of the variables used in the split at each, `index`, the cut-off value of splitting variable for binary partitioning, `p (Instability)`, the p-value for parameter instability test for the splitting variable, `loglik`, the log-likelihood of the node, `improve`, the improvement in deviance given by this split, and `Terminal`, indicator (True or False) of terminal node.

`p` number of fixed parameters

`AIC.tree` AIC of the tree-structured model

`AIC.root` AIC at the root node (i.e., without tree structure)

`improve.AIC` improvement in AIC due to tree structure (AIC.tree - AIC.root)

`logLik.tree` log-likelihood of the tree-structured model

`logLik.root` log-likelihood at the root node (i.e., without tree structure)

`Deviance` 2*(logLik.tree-logLik.root)

`LRT.df` degrees of freedom for likelihood ratio test comparing tree-structured model with the model at root node.

`LRT.p` p-value for likelihood ratio test comparing tree-structured model with the model at root node.

`subj.class` Assigned node for each individual subjects per fitted longitudinal tree

`frame` rpart compatible object

`splits` rpart compatible object

`cptable` rpart compatible object

`functions` rpart compatible object

Author(s)

Madan Gopal Kundu <`madan_g.kundu@yahoo.com`>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. Biostatistics & Epidemiology, 3(1):1-22.

See Also

`plotLongCART`, `textLongCART`, `StabCat`, `StabCont`

Examples

```
#--- Get the data
data(ACTG175)

#--- Run LongCART()
gvars=c("gender", "wtkg", "hemo", "homo", "drugs",
        "karnof", "oprior", "z30", "zprior", "race",
        "str2", "symptom", "treat", "offtrt")
tgvars=c(0, 1, 0, 0, 0,
         1, 0, 0, 0, 0,
         0, 0, 0, 0)


out<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#--- Plot tree

par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

plot.SurvCART *Plot an SurvCART Object*

Description

Plots an SurvCART object on the current graphics device.

Usage

```
## S3 method for class 'SurvCART'
plot(x, uniform = FALSE, branch = 1, compress = FALSE,
              nspace = branch, margin = 0, minbranch = 0.3, ...)
```

Arguments

`x` a fitted object of class `"SurvCART"`, containing a survival tree.

`uniform` similar to `plot.rpart`; if TRUE, uniform vertical spacing of the nodes is used; this may be less cluttered when fitting a large plot onto a page. The default is to use a non-uniform spacing proportional to the error in the fit.

`branch` similar to `plot.rpart`; controls the shape of the branches from parent to child node. Any number from 0 to 1 is allowed. A value of 1 gives square shouldered branches, a value of 0 give V shaped branches, with other values being intermediate.

`compress` similar to `plot.rpart`; if `FALSE`, the leaf nodes will be at the horizontal plot coordinates of `1:nleaves`. If `TRUE`, the routine attempts a more compact arrangement of the tree.

`nspace` similar to `plot.rpart`; the amount of extra space between a node with children and a leaf, as compared to the minimal space between leaves. Applies to compressed trees only. The default is the value of `branch`.

`margin` similar to `plot.rpart`; an extra fraction of white space to leave around the borders of the tree. (Long labels sometimes get cut off by the default computation).

`minbranch` similar to `plot.rpart`; set the minimum length for a branch to `minbranch` times the average branch length. This parameter is ignored if `uniform=TRUE`. Sometimes a split will give very little improvement, or even (in the classification case) no improvement at all. A tree with branch lengths strictly proportional to improvement leaves no room to squeeze in node labels.

`...` arguments to be passed to or from other methods.

Details

This function is a method for the generic function plot, for objects of class `SurvCART`. The y-coordinate of the top node of the tree will always be 1.

Value

The coordinates of the nodes are returned as a list, with components `x` and `y`.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

text.SurvCART, SurvCART, KMPlot.SurvCART, StabCat.surv, StabCont.surv

Examples

```
#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART()
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time", event.ind=1,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
        alpha=0.05, minsplit=80,
        minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

plotLongCART *Plot an LongCART Object*

Description

Plots an LongCART object on the current graphics device.

Usage

```
plotLongCART(x, uniform = FALSE, branch = 1, nspace = branch,
     margin = 0, minbranch = 0.3, ...)
```

Arguments

`x` a fitted object of class `"LongCART"`, containing a linear mixed effects tree.

`uniform` similar to `plot.rpart`; if TRUE, uniform vertical spacing of the nodes is used; this may be less cluttered when fitting a large plot onto a page. The default is to use a non-uniform spacing proportional to the error in the fit.

`branch` similar to `plot.rpart`; controls the shape of the branches from parent to child node. Any number from 0 to 1 is allowed. A value of 1 gives square shouldered branches, a value of 0 give V shaped branches, with other values being intermediate.

`nspace` similar to `plot.rpart`; the amount of extra space between a node with children and a leaf, as compared to the minimal space between leaves. Applies to compressed trees only. The default is the value of `branch`.

`margin` similar to `plot.rpart`; an extra fraction of white space to leave around the borders of the tree. (Long labels sometimes get cut off by the default computation).

`minbranch` similar to `plot.rpart`; set the minimum length for a branch to `minbranch` times the average branch length. This parameter is ignored if `uniform=TRUE`. Sometimes a split will give very little improvement, or even (in the classification case) no improvement at all. A tree with branch lengths strictly proportional to improvement leaves no room to squeeze in node labels.

`...` arguments to be passed to or from other methods.

Details

This function is a method for the generic function plot, for objects of class `LongCART`. The y-coordinate of the top node of the tree will always be 1.

Value

The coordinates of the nodes are returned as a list, with components `x` and `y`.

Author(s)

Madan Gopal Kundu `<madan_g.kundu@yahoo.com>`

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. Biostatistics & Epidemiology, 3(1):1-22.

See Also

`textLongCART`, `LongCART`, `StabCat`, `StabCont`

Examples

```
#--- Get the data
data(ACTG175)

#--- Run LongCART()
gvars=c("gender", "wtkg", "hemo", "homo", "drugs",
        "karnof", "oprior", "z30", "zprior", "race",
        "str2", "symptom", "treat", "offtrt")
tgvars=c(0, 1, 0, 0, 0,
         1, 0, 0, 0, 0,
         0, 0, 0, 0)

out<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

StabCat *parameter stability test for categorical partitioning variable*

Description

Performs parameter stability test (Kundu and Harezlak, 2019) with categorical partitioning variable to determine whether the parameters of linear mixed effects model remains same across all distinct values of given categorical partitioning variable.

Usage

```
StabCat(data, patid, fixed, splitvar)
```

Arguments

`data` name of the dataset. It must contain variable specified for `patid` (indicating subject id) and all the variables specified in the formula and the caterogrical partitioning variable of interest specified in `splitvar`. Note that, only numerically coded categorical variable should be specified.

`patid` name of the subject id variable.

`fixed` a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Model with `-1` to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only `1` right to the ~ operator.

`splitvar` the categorical partitioning variable of interest. It's value should not change over time.

Details

The categorical partitioning variable of interest. It's value should not change over time.

`Y_i(t)= W_i(t) theta + b_i + epsilon_{it}`

where `W_i(t)` is the design matrix, `theta` is the parameter associated with `W_i(t)` and `b_i` is the random intercept. Also, `epsilon_{it} ~ N(0,sigma ^2)` and `b_i ~ N(0,sigma_u^2)`. Let X be the baseline categorical partitioning variable of interest. `StabCat()` performs the following omnibus test

`H_0:theta_{(g)}=theta_0` vs. `H_1: theta_{(g)} ^= theta_0`, for all g

where, `theta_{(g)}` is the true value of `theta` for subjects with `X=C_g` where `C_g` is the any value realized by `X`.

Value

`p` It returns the p-value for parameter instability test

Author(s)

Madan Gopal Kundu `<madan_g.kundu@yahoo.com>`

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. Biostatistics & Epidemiology, 3(1):1-22.

See Also

`StabCont`, `LongCART`, `plotLongCART`, `textLongCART`

Examples

```
#--- Get the data
data(ACTG175)

#--- Run StabCat()
out<- StabCat(data=ACTG175, patid="pidnum", fixed=cd4~time, splitvar="gender")
out$pval
```

`StabCat.surv` *parameter stability test for categorical partitioning variable*

Description

Performs parameter stability test (Kundu, 2020) with categorical partitioning variable to determine whether the parameters of exponential time-to-event distribution and exponential censoring distribution remain same across all distinct values of given categorical partitioning variable.

Usage

```
StabCat.surv(data, timevar, censorvar, splitvar,
             time.dist="exponential", cens.dist="NA", event.ind=1, print=FALSE)
```

Arguments

`data` name of the dataset. It must contain variable specified for `timevar` (indicating follow-up times), `censorvar` (indicating censoring status) and the caterogrical partitioning variable of interest specified in `splitvar`. Note that, only numerically coded categorical variable should be specified.

`timevar` name of the variable with follow-up times.

`censorvar` name of the variable with censoring status.

`time.dist` name of time-to-event distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"` or `"normal"`.

`cens.dist` name of censoring distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"`, `"normal"` or `"NA"`. If specified `"NA"`, then parameter instability test corresponding to censoring distribution will not be performed.

`event.ind` value of the censoring variable indicating event.

`splitvar` the categorical partitioning variable of interest. It's value should not change over time.

`print` if `TRUE`, then additional information including estimated parameters, score function and its variance will be printed.

Details

`StabCat.surv()` performs the following omnibus test

`H_0:lambda_{(g)}=lambda_0` vs. `H_1: lambda_{(g)} ^= lambda_0`, for all g

where, `theta_{(g)}` is the true value of `theta` for subjects with `X=C_g`. `theta` includes all the parameters of time to event distribution and also parameters of censoring distribution, if specified. `C_g` is the any value realized by categorical partitioning variable `X`.

Exponential distribution: f(t)=lambda*exp(-lambda*t)

Weibull distribution: f(t)=alpha*lambda*t^(alpha-1)*exp(-lambda*t^alpha)

Lognormal distribution: f(t)=(1/t)*(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(log(t)-mu)/sigma^2]

Normal distribution: f(t)=(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(t-mu)/sigma^2]

Value

pval
: p-value for parameter instability test

type
: 1, if event times are more heterogeneous; 2, if censoring times are more heterogeneous.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

StabCont.surv, SurvCART, plot.SurvCART, text.SurvCART

Examples

```
#--- time-to-event distribution: exponential, censoring distribution: None
out1<- StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex", event.ind=2)
out1$pval

#--- time-to-event distribution: weibull, censoring distribution: None
StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex",
             time.dist="weibull", event.ind=2)

#--- time-to-event distribution: weibull, censoring distribution: exponential
StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex",
             time.dist="weibull", cens.dist="exponential", event.ind=2)
```

StabCont *parameter stability test for continuous partitioning variable*

Description

Performs parameter stability test (Kundu and Harezlak, 2019) with continuous partitioning variable to determine whether the parameters of linear mixed effects model remains same across all distinct values of given continuous partitioning variable.

Usage

```
StabCont(data, patid, fixed, splitvar)
```

Arguments

`data` name of the dataset. It must contain variable specified for `patid` (indicating subject id) and all the variables specified in the formula and the StabCont(data, fixed, splitvar)partitioning variable of interest specified in `splitvar`.

`patid` name of the subject id variable.

`fixed` a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Model with `-1` to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only `1` right to the ~ operator.

`splitvar` the continuous partitioning variable of interest. It's value should not change over time.

Details

The continuous partitioning variable of interest. It's value should not change over time.

`Y_i(t)= W_i(t)theta + b_i + epsilon_{it}`

where `W_i(t)` is the design matrix, `theta` is the parameter associated with `W_i(t)` and `b_i` is the random intercept. Also, `epsilon_{it} ~ N(0,sigma ^2)` and `b_i ~ N(0,sigma_u^2)`. Let `X` be the baseline continuous partitioning variable of interest. `StabCont()` performs the following omnibus test

`H_0:theta_{(g)}=theta_0` vs. `H_1: theta_{(g)} ^= theta_0`, for all g

where, `theta_{(g)}` is the true value of `theta` for subjects with `X=C_g` where `C_g` is the any value realized by `X`.

Value

`p` It returns the p-value for parameter instability test

Author(s)

Madan Gopal Kundu `<madan_g.kundu@yahoo.com>`

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. Biostatistics & Epidemiology, 3(1):1-22.

See Also

`StabCont`, `LongCART`, `plotLongCART`, `textLongCART`

Examples

```
#--- Get the data
data(ACTG175)

#--- Run StabCont()
out<- StabCont(data=ACTG175, patid="pidnum", fixed=cd4~time, splitvar="age")
out$pval
```

`StabCont.surv` *parameter stability test for continuous partitioning variable*

Description

Performs parameter stability test (Kundu, 2020) with continuous partitioning variable to determine whether the parameters of exponential time-to-event distribution and exponential censoring distribution remain same across all distinct values of given continupus partitioning variable.

Usage

```
StabCont.surv(data, timevar, censorvar, splitvar,
             time.dist="exponential", cens.dist="NA", event.ind=1, print=FALSE)
```

Arguments

`data` name of the dataset. It must contain variable specified for `timevar` (indicating follow-up times), `censorvar` (indicating censoring status) and the caterogrical partitioning variable of interest specified in `splitvar`. Note that, only numerically coded categorical variable should be specified.

`timevar` name of the variable with follow-up times.

`censorvar` name of the variable with censoring status.

`time.dist` name of time-to-event distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"` or `"normal"`.

`cens.dist` name of censoring distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"`, `"normal"` or `"NA"`. If specified `"NA"`, then parameter instability test corresponding to censoring distribution will not be performed.

`event.ind` value of the censoring variable indicating event.

`splitvar` the continuous partitioning variable of interest.

`print` if `TRUE`, then additional information including estimated parameters, score function and its variance will be printed.

Details

`StabCont.surv()` performs the following omnibus test

`H_0:theta_{(g)}=theta_0` vs. `H_1: theta_{(g)} ^= theta_0`, for all g

where, `theta_{(g)}` is the true value of `theta` for subjects with `X=C_g`. `theta` includes all the parameters of time to event distribution and also parameters of censoring distribution, if specified. `C_g` is the any value realized by continuous partitioning variable `X`.

Exponential distribution: f(t)=lambda*exp(-lambda*t)

Weibull distribution: f(t)=alpha*lambda*t^(alpha-1)*exp(-lambda*t^alpha)

Lognormal distribution: f(t)=(1/t)*(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(log(t)-mu)/sigma^2]

Normal distribution: f(t)=(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(t-mu)/sigma^2]

Value

`pval` p-value for parameter instability test

`type` 1, if event times are more heterogeneous; 2, if censoring times are more heterogeneous.

Author(s)

Madan Gopal Kundu <`madan_g.kundu@yahoo.com`>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

`StabCont.surv`, `SurvCART`, `plot.SurvCART`, `text.SurvCART`

Examples

```
#--- time-to-event distribution: exponential, censoring distribution: None
out1<- StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age", event.ind=2)
out1$pval

#--- time-to-event distribution: weibull, censoring distribution: None
StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age",
              time.dist="weibull", event.ind=2)

#--- time-to-event distribution: weibull, censoring distribution: exponential
StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age",
              time.dist="weibull", cens.dist="exponential", event.ind=2)
```

SurvCART *Survival CART with time to event response via binary partitioning*

Description

Recursive partitioning for linear mixed effects model with survival data per SurvCART algorithm based on baseline partitioning variables (Kundu, 2020).

Usage

```
SurvCART(data, patid, timevar, censorvar, gvars, tgvars,
         time.dist="exponential", cens.dist="NA", event.ind=1,
         alpha=0.05, minsplit=40, minbucket=20, print=FALSE)
```

Arguments

`data`	name of the dataset. It must contain variable specified for `patid` (indicating subject id), all the variables specified in the formula and the baseline partitioning variables.
`patid`	name of the subject id variable.
`timevar`	name of the variable with follow-up times.
`censorvar`	name of the variable with censoring status.
`gvars`	list of partitioning variables of interest. Value of these variables should not change over time. Regarding categorical variables, only numerically coded categorical variables should be specified. For nominal categorical variables or factors, please first create corresponding dummy variable(s) and then pass through `gvars`.
`tgvars`	types (categorical or continuous) of partitioning variables specified in `gvar`. For each of continuous partitioning variables, specify `0` and for each of the categorical partitioning variables, specify `1`. Length of `tgvars` should match to the length of `gvars`
`time.dist`	name of time-to-event distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"` or `"normal"`.
`cens.dist`	name of censoring distribution. It can be one of the following distributions: `"exponential"`, `"weibull"`, `"lognormal"`, `"normal"` or `"NA"`. If specified `"NA"`, then parameter instability test corresponding to censoring distribution will not be performed.
`event.ind`	value of the censoring variable indicating event.
`alpha`	alpha (i.e., nominal type I error) level for parameter instability test
`minsplit`	the minimum number of observations that must exist in a node in order for a split to be attempted.
`minbucket`	he minimum number of observations in any terminal node.
`print`	if `TRUE`, then summary such as number of subjects at risk, number of events, median event time and median censoring time model will be printed for each node.

Details

Construct survival tree based on heterogeneity in time-to-event and censoring distributions.

Exponential distribution: f(t)=lambda*exp(-lambda*t)

Weibull distribution: f(t)=alpha*lambda*t^(alpha-1)*exp(-lambda*t^alpha)

Lognormal distribution: f(t)=(1/t)*(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(log(t)-mu)/sigma^2]

Normal distribution: f(t)=(1/sqrt(2*pi*sigma^2))*exp[-(1/2)*(t-mu)/sigma^2]

Value

`Treeout` contains summary information of tree fitting for each terminal nodes and non-terminal nodes. Columns of `Treeout` include "ID", the (unique) node numbers that follow a binary ordering indexed by node depth, `n`, the number of subject-sreaching the node, `D`, the number of events reaching the node, `median.T`, the median survival time at the node, `median.C`, the median censoring time at the node, var, splitting variable, `index`, the cut-off value of splitting variable for binary partitioning, `p (Instability)`, the p-value for parameter instability test for the splitting variable, `loglik`, the log-likelihood of the node, `improve`, the improvement in deviance given by this split, and `Terminal`, indicator (True or False) of terminal node.

`logLik.tree` log-likelihood of the tree-structured model, based on Cox model including sub-groups as covariates

`logLik.root` log-likelihood at the root node (i.e., without tree structure), based on Cox model without any covariate

`AIC.tree` AIC of the tree-structured model, based on Cox model including sub-groups as covariates

`AIC.root` AIC at the root node (i.e., without tree structure), based on Cox model without any covariate

`subj.class` Assigned node for each individual subjects per fitted longitudinal tree

`frame` rpart compatible object

`splits` rpart compatible object

`cptable` rpart compatible object

`functions` rpart compatible object

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

`plot.SurvCART`, `KMPlot.SurvCART`, `text.SurvCART`, `StabCat.surv`, `StabCont.surv`

Examples

```
#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART() with time-to-event distribution: exponential, censoring distribution: None
out<- SurvCART(data=GBSG2, patid=”subjid”, censorvar=”cens”, timevar=”time”,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
        event.ind=1,  alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)

#Plot KM plot for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)


#--- Run SurvCART() with time-to-event distribution: weibull censoring distribution: None
out2<- SurvCART(data=GBSG2, patid=”subjid”, censorvar=”cens”, timevar=”time”,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
     time.dist=”weibull”, event.ind=1, alpha=0.05, minsplit=80, minbucket=40, print=TRUE)


#--- Run SurvCART() with time-to-event distribution: weibull censoring distribution: exponential
out<- SurvCART(data=GBSG2, patid=”subjid”, censorvar=”cens”, timevar=”time”,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
        time.dist=”weibull”, cens.dist=”exponential”, event.ind=1,
        alpha=0.05, minsplit=80, minbucket=40, print=TRUE)
```

`text.SurvCART` *Place text on SurvCART tree*

Description

Labels the current plot of the tree generated from SurvCART object with text.

Usage

```
## S3 method for class 'SurvCART'
text(x, splits = TRUE, all = FALSE,
             use.n = FALSE, minlength = 1L, ...)
```

Arguments

`x` a fitted object of class "`SurvCART`", containing a survival tree.

`splits` similar to text.rpart; logical flag. If `TRUE` (default), then the splits in the tree are labeled with the criterion for the split.

`all` similar to text.rpart; Logical. If TRUE, all nodes are labeled, otherwise just terminal nodes.

`use.n` similar to text.rpart; Logical. If TRUE, adds `n` to label.

`minlength` similar to text.rpart; the length to use for factor labels. A value of 1 causes them to be printed as ’a’, ’b’, ... Larger values use abbreviations of the label names. See the labels.rpart function for details.

`...` arguments to be passed to or from other methods.

Author(s)

Madan Gopal Kundu <`madan_g.kundu@yahoo.com`>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

`plot.SurvCART`, `SurvCART`, `KMPlot.SurvCART`, `StabCat.surv`, `StabCont.surv`

Examples

```
#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART()
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time", event.ind=1,
     gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
        tgvars=c(0,1,0,1,0,1, 1,1),
        alpha=0.05, minsplit=80,
        minbucket=40, print=TRUE)
```

```
#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

textLongCART *Place text on LongCART tree*

Description

Labels the current plot of the tree generated from LongCART object with text.

Usage

```
textLongCART(x, splits = TRUE, all = FALSE,
             use.n = FALSE, minlength = 1L, ...)
```

Arguments

`x` a fitted object of class `"LongCART"`, containing a linear mixed effects tree.

`splits` similar to plot.rpart; logical flag. If `TRUE` (default), then the splits in the tree are labeled with the criterion for the split.

`all` similar to plot.rpart; Logical. If TRUE, all nodes are labeled, otherwise just terminal nodes.

`use.n` similar to plot.rpart; Logical. If TRUE, adds `n` to label.

`minlength` the length to use for factor labels. A value of 1 causes them to be printed as 'a', 'b', ... Larger values use abbreviations of the label names. See the labels.rpart function for details.

`...` arguments to be passed to or from other methods.

Author(s)

Madan Gopal Kundu <`madan_g.kundu@yahoo.com`>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. Biostatistics & Epidemiology, 3(1):1-22.

See Also

`plotLongCART`, `LongCART`, `StabCat`, `StabCont`

Examples

```
#--- Get the data
data(ACTG175)

#--- Run LongCART()
gvars=c("gender", "wtkg", "hemo", "homo", "drugs",
        "karnof", "oprior", "z30", "zprior", "race",
        "str2", "symptom", "treat", "offtrt")
tgvars=c(0, 1, 0, 0, 0,
         1, 0, 0, 0, 0,
         0, 0, 0, 0)

out<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

Index

ACTG175, 2

GBSG2, 3

KMPlot.SurvCART, 4, *9*, *19*, *21*

LongCART, 5, *10*, *12*, *15*, *22*

plot *(*plot.SurvCART*)*, 8
plot.SurvCART, *4*, 8, *14*, *17*, *19*, *21*
plotLongCART, *7*, 9, *12*, *15*, *22*

StabCat, *7*, *10*, 11, *22*
StabCat *(*StabCat*)*, 11
StabCat.surv, *4*, *9*, 13, *19*, *21*
StabCat.surv *(*StabCat.surv*)*, 13
StabCont, *7*, *10*, *12*, 14, *15*, *22*
StabCont *(*StabCont*)*, 14
StabCont.surv, *4*, *9*, *14*, 16, *17*, *19*, *21*
StabCont.surv *(*StabCont.surv*)*, 16
SurvCART, *4*, *9*, *14*, *17*, 18, *21*

text *(*text.SurvCART*)*, 20
text.SurvCART, *4*, *9*, *14*, *17*, *19*, 20
textLongCART, *7*, *10*, *12*, *15*, 22