Package ‘LTRCtrees’

October 12, 2022

Type Package

Title Survival Trees to Fit Left-Truncated and Right-Censored and
Interval-Censored Survival Data

Version 1.1.1

Description Recursive partition algorithms designed for fitting survival tree with left-
truncated and right censored (LTRC) data, as well as interval-censored data.
The LTRC trees can also be used to fit survival tree with time-varying covariates.

Imports partykit (>= 1.2.0), rpart, survival, inum, icenReg, Icens, interval

biocViews

Suggests Formula, rpart.plot, knitr, rmarkdown

Depends R (>= 3.2.0)

License GPL-3

LazyData TRUE

Author Wei Fu, Jeffrey Simonoff, Wenbo Jing

Maintainer Wenbo Jing <wj2093@stern.nyu.edu>

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-13 13:00:02 UTC

R topics documented:

.. graphviz-doc::

   .. literalinclude:: ./trees.r
   :language: r

- .logrank_trafo2
- extree_data
- Ictree
- LTRCART
- LTRCIT
- Pred.rpart

Index

11
.logrank_trafo2  
*Logrank transformation function for LTRC data*

**Description**

.logrank_trafo transforms `Surv(time1, time2, event)` objects into logrank scores, which will be used later in the tree algorithm. It is not designed to be used by users, not for internal used of LTRCIT function.

**Usage**

`.logrank_trafo2(x2)`

**Arguments**

- `x2`  
  A vector `Surv(Surv(time1, time2, event))` objects

**Value**

Logrank scores of LTRC objects

---

extree_data  
*Copy the partykit::extree_data function from partykit to avoid dependency issue*

**Description**

extree_data imports partykit::extree_data function

**Usage**

`extree_data(`

- `formula`,
- `data`,
- `subset`,
- `na.action = stats::na.pass`,
- `weights`,
- `offset`,
- `cluster`,
- `strata`,
- `scores = NULL`,
- `yx = c("none", "matrix")`,
- `ytype = c("vector", "data.frame", "matrix")`,
- `nmax = c(yx = Inf, z = Inf)`,
- `...`  
  `)`
ICtree

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>data</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>subset</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>na.action</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>weights</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>offset</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>cluster</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>strata</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>scores</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>yx</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>ytype</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>nmax</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
<tr>
<td>...</td>
<td>Same as the one in extree_data, check extree_data for usage</td>
</tr>
</tbody>
</table>

Value

check extree_data for the return value

ICtree

Fit a survival tree for interval-censored survival data

Description

Recursive partition for interval-censored survival data in a conditional inference framework.

Usage

ICtree(Formula, data, Control = partykit::ctree_control())

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>A formula object, with the response be a Surv object, with form Surv(time1, time2, type=&quot;interval2&quot;)</td>
</tr>
<tr>
<td>data</td>
<td>A data frame contains the variables named in Formula.</td>
</tr>
<tr>
<td>Control</td>
<td>A list of control parameters, see ctree_control</td>
</tr>
</tbody>
</table>
Details

ICtree returns a party object. This function extends the conditional inference survival tree algorithm in ctree to fit interval-censored survival data. This function itself not longer requires the interval package, but running the example below requires the interval package (for bcos data), which in turn requires the Icens package, which is not available on CRAN. To install the Icens package, enter the following commands

source("https://bioconductor.org/biocLite.R")
biocLite("Icens")

Value

An object of class party.

References


Examples

library(Icens)
library(interval)
library(LTRCtrees)
data(bcos)

## Fit ICtree survival tree
## make sure to attach survival package (by library(survival) ) before using Surv function
Ctree <- ICtree(Surv(left,right,type="interval2")~treatment, data = bcos)

## Plot the fitted tree
plot(Ctree)

LTRCART

Fit a relative risk survival tree for LTRC data

Description

LTRCART returns an rpart object. This function extends the survival tree algorithm in rpart to fit left-truncated and right censored (LTRC) data.

Usage

LTRCART(
  formula,
  data,
  weights = NULL,
subset = NULL,
no.SE = 0,
control = rpart::rpart.control(cp = 0.001)
)

Arguments

formula A formula object specifies the regression function, with the response be a Surv object, with form Surv(time1, time2, event)
data An optional data frame which contains the variables named in the formula.weights Optional case weights, same as in rpartsubset Optional expression saying that only a subset of the rows of the data should be used in the fit, same as in rpartno.SE Number of standard errors used in pruning, with default value 0.control A list of control values used to control the rpart algorithm, with default cp = 0.001. See rpart.control for details.

Value

An object of class rpart. See rpart.object.

References


Examples

## The Assay of serum free light chain data in survival package
## Adjust data & clean data
library(survival)
library(LTRCtrees)
Data <- flchain
Data <- Data[!is.na(Data$creatinine),]
Data$End <- Data$age + Data$futime/365
DATA <- Data[Data$End > Data$age,]
names(DATA)[6] <- "FLC"

## Setup training set and test set
Train = DATA[1:500,]
Test = DATA[1000:1020,]

## Fit LTRCART survival tree
## make sure to attach survival package (by library(survival) ) before using Surv function
LTRCART.obj <- LTRCART(Surv(age, End, death) ~ sex + FLC + creatinine, Train)

## Putting Surv(End, death) in formula would result an error message
## since LTRCART is expecting Surv(time1, time2, event)

## Plot the fitted tree
library(rpart.plot)
rpart.plot(LTRCART.obj)

## Plot as partykit::party object
library(partykit)
plot(as.party(LTRCART.obj))

## Plot as partykit::party object with survival curves on terminal nodes
LTRCART.obj.party <- as.party(LTRCART.obj)
LTRCART.obj.party$fitted[["(response)"]] <- Surv(Train$age, Train$End, Train$death)
plot(LTRCART.obj.party)

## Predict relative risk on test set
LTRCART.pred <- predict(LTRCART.obj, newdata = Test)

###########################################################################
### Survival tree with time-varying covariates ##########################
###########################################################################
## The pbcseq dataset of survival package
library(survival)
## Create the start-stop-event triplet needed for coxph and LTRCART trees
first <- with(pbcseq, c(TRUE, diff(id) !=0)) #first id for each subject
last <- c(first[-1], TRUE) #last id
time1 <- with(pbcseq, ifelse(first, 0, day))
time2 <- with(pbcseq, ifelse(last, futime, c(day[-1], 0)))
event <- with(pbcseq, ifelse(last, status, 0))
event <- 1*(event==2)
pbcseq$time1 <- time1
pbcseq$time2 <- time2
pbcseq$event <- event

## Fit the Cox model and LTRCART tree with time-varying covariates
fit.cox <- coxph(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
LTRCART.fit <- LTRCART(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
rpart.plot(LTRCART.fit)

### transform the wide format data into long format data using tmerge function
### from survival function
### Stanford Heart Transplant data
jasa$subject <- 1:nrow(jasa)
tdata <- with(jasa, data.frame(subject = subject,
                               futime= pmax(.5, fu.date - accept.dt),
                               txtime= ifelse(tx.date== fu.date,
                               (tx.date -accept.dt) -.5,
                               (tx.date - accept.dt)),
                               fustat = fustat))

sdata <- tmerge(jasa, tdata, id=subject,death = event(futime, fustat),
                trt = tdc(txtime), options= list(idname="subject"))
sdata$age <- sdata$age - 48
sdata$year <- as.numeric(sdata$accept.dt - as.Date("1967-10-01"))/365.25

Cox.fit <- coxph(Surv(tstart, tstop, death) ~ age + surgery, data = sdata)
LTRCART.fit <- LTRCART(Surv(tstart, tstop, death) ~ age + transplant, data = sdata)
rpart.plot(LTRCART.fit)

---

**LTRCIT**

*Fit a conditional inference survival tree for LTRC data*

**Description**

LTRCIT returns a party object. This function extends the conditional inference survival tree algorithm in ctree to fit left-truncated and right censored (LTRC) data.

**Usage**

LTRCIT(Formula, data, Control = partykit::ctree_control())

**Arguments**

- **Formula**: A formula object, with the response be a Surv object, with form Surv(time1, time2, event)
- **data**: A data frame contains the variables named in formula.
- **Control**: A list of control parameters, see ctree_control

**Value**

An object of class party.

**References**


**Examples**

```r
## The Assay of serum free light chain data in survival package
## Adjust data & clean data
library(survival)
library(LTRCtrees)
Data <- flchain
Data <- Data[!is.na(Data$creatinine),]
Data$End <- Data$age + Data$futime/365
DATA <- Data[Data$End > Data$age,]
names(DATA)[6] <- "FLC"
```
## Setup training set and test set
Train = DATA[1:500,]
Test = DATA[1000:1020,]

## Fit LTRCIT survival tree
## make sure to attach survival package (by library(survival) ) before using Surv function
LTRCIT.obj <- LTRCIT(Surv(age, End, death) ~ sex + FLC + creatinine, Train)
plot(LTRCIT.obj)

## Putting Surv(End, death) in formula would result an error message
## since LTRCIT is expecting Surv(time1, time2, event)

## Note that LTRCIT.obj is an object of class party
## predict median survival time on test data
LTRCIT.pred <- predict(LTRCIT.obj, newdata = Test, type = "response")

## predict Kaplan Meier survival curve on test data,
## return a list of survfit objects -- the predicted KM curves
LTRCIT.pred <- predict(LTRCIT.obj, newdata = Test, type = "prob")

### Survival tree with time-varying covariates

## The pbcseq dataset of survival package
library(survival)
## Create the start-stop-event triplet needed for coxph and LTRC trees
first <- with(pbcseq, c(TRUE, diff(id) !=0)) #first id for each subject
last <- c(first[-1], TRUE) #last id
time1 <- with(pbcseq, ifelse(first, 0, day))
time2 <- with(pbcseq, ifelse(last, futime, c(day[-1], 0)))
event <- with(pbcseq, ifelse(last, status, 0))
event <- 1*(event==2)
pbcseq$time1 <- time1
pbcseq$time2 <- time2
pbcseq$event <- event

pbcseq = pbcseq[1:1000,] ## fit on subset of the data to save fitting time
## Fit the Cox model and LTRCIT tree with time-varying covariates
fit.cox <- coxph(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
LTRCIT.fit <- LTRCIT(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
plot(LTRCIT.fit)

## transform the wide format data into long format data using tmerge function
## from survival function
## Stanford Heart Transplant data
jasa$subject <- 1:nrow(jasa)

tdata <- with(jasa, data.frame(subject = subject,
    futime= pmax(.5, fu.date - accept.dt),
    txtime= ifelse(tx.date== fu.date,
        (tx.date - accept.dt) -.5,
        (tx.date - accept.dt)),
    event= 1*(event==2))

jasa$subject <- 1:nrow(jasa)

## Fit the Cox model and LTRCIT tree with time-varying covariates
fit.cox <- coxph(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
LTRCIT.fit <- LTRCIT(Surv(time1, time2, event) ~ age + sex + log(bili), pbcseq)
plot(LTRCIT.fit)
fustat = fustat))

sdata <- tmerge(jasa, tdata, id=subject, death = event(futime, fustat),
  trt = tdc(txtime), options= list(idname="subject"))

sdata$age <- sdata$age - 48

sdata$year <- as.numeric(sdata$accept.dt - as.Date("1967-10-01"))/365.25

Cox.fit <- coxph(Surv(tstart, tstop, death) ~ age+ surgery, data= sdata)
LTRCIT.fit <- LTRCIT(Surv(tstart, tstop, death) ~ age + transplant, data = sdata)
plot(LTRCIT.fit)

---

**Pred.rpart**

*Prediction function for rpart.object*

**Description**

The output of LTRCART is an *rpart* object, and as a result the usual *predict* function on such an object returns the predicted relative risk on the test set. *Pred.rpart* returns the predicted Kaplan-Meier curves and median survival times on the test set, which in some circumstances might be desirable in practice. Note that this function can be applied to any *rpart* survival tree object, not just one produced by LTRCART

**Usage**

`Pred.rpart(formula, train, test)`

**Arguments**

- `formula` A formula used to fit the survival tree. The response is a *Surv* object. If it has the form `Surv(time1, time2, event)`, then LTRCART is called internally; if response has the form `Surv(time, event)`, then the *rpart* is called internally.
- `train` Training set
- `test` Test set

**Value**

A list of predicted KM curves and median survival times.

**Examples**

```
## The Assay of serum free light chain data in survival package
## Adjust data & clean data
library(survival)
library(LTRCtrees)
Data <- flchain
```
Data <- Data[!is.na(Data$creatinine),]
Data$End <- Data$age + Data$futime/365
DATA <- Data[Data$End > Data$age,]
names(DATA)[6] <- "FLC"

## Setup training set and test set
Train = DATA[1:500,]
Test = DATA[1000:1020,]

## Predict median survival time and Kaplan Meier survival curve
## on test data using Pred.rpart
LTRCART.pred <- Pred.rpart(Surv(age, End, death) ~ sex + FLC + creatinine, Train, Test)
LTRCART.pred$KMcurves  ## list of predicted KM curves
LTRCART.pred$Medians  ## vector of predicted median survival time
Index

.logrank_trafo2, 2
ctree, 4, 7
ctree_control, 3, 7
extree_data, 2, 3
ICtree, 3
LTRCART, 4
LTRCIT, 7
party, 4, 7
Pred.rpart, 9
predict, 9
rpart, 4, 5, 9
rpart.control, 5
rpart.object, 5, 9
Surv, 2, 3, 5, 7, 9