Package 'CureAuxSP'

February 28, 2024

Title Mixture Cure Models with Auxiliary Subgroup Survival Probabilities

Version 0.0.1

Author Jie Ding [aut, cre] (<https://orcid.org/0000-0002-6083-7529>), Jialiang Li [aut] (<https://orcid.org/0000-0002-9704-4135>), Mengxiu Zhang [aut] (<https://orcid.org/0009-0001-3773-3019>), Xiaoguang Wang [aut] (<https://orcid.org/0000-0002-5598-062X>)

Maintainer Jie Ding <dingjie@mail.dlut.edu.cn>

Description Estimate mixture cure models with subgroup survival probabilities as auxiliary information. A reference of the underlying methods is Jie Ding, Jialiang Li and Xiaoguang Wang (2024) <doi:10.1093/jrsssc/qlad106>.

License GPL (>= 2)

Encoding UTF-8

Imports MASS, survival, lars, mvtnorm, methods, TCGAbiolinks

RoxygenNote 7.2.3

URL <https://github.com/biostat-jieding/CureAuxSP>

NeedsCompilation no

Repository CRAN

Date/Publication 2024-02-28 18:50:02 UTC

R topics documented:

`CureAuxSP` *CureAuxSP: Mixture Cure Models with Auxiliary Subgroup Survival Probabilities.*

Description

This package provides an information synthesis framework that can Estimate mixture cure models with subgroup survival probabilities as auxiliary information. The underlying methods are based on the paper titled "Efficient auxiliary information synthesis for cure rate model", which has been published in Jie Ding, Jialiang Li and Xiaoguang Wang (2024) doi:10.1093/jrsssc/qlad106. project work

`Print.SMC.AuxSP` *Print SMC.AuxSP object returned by our main function SMC.AuxSP()*

Description

Output of `SMC.AuxSP` object.

Usage

```
Print.SMC.AuxSP(object)
```

Arguments

`object`	an object of SMC.AuxSP

Value

This function has no return value and is used to print the results in a `SMC.AuxSP` class with a better presentation.

`Probs.Sub` *Calculate subgroup survival probabilities*

Description

Calculate Subgroup survival probabilities basedon the Kaplan-Meier estimation procedure

Usage

```
Probs.Sub(tstar, sdata, G)
```

Arguments

tstar	time points that the survival probabilities will be estimated at.
sdata	a survival dataset (dataframe) in which to interpret the variables named in the formula and the cureform.
G	a matrix used to indicate which subgroups he/she belongs to for each of these subjects.

Value

It returns the estimated subgroup survival probabilities for a given survival dataset.

---

sdata.SMC *Generate simulated dataset from a well-designed PH mixture cure model*

Description

Generate simulated dataset from a well-designed PH mixture cure model.

Usage

```
sdata.SMC(n, trace = FALSE)
```

Arguments

n	the sample size of the simulated dataset.
trace	a logical value that indicates whether the information about cure rate and censoring rate should be printed.

Value

It returns the simulated dataset from a well-designed PH mixture cure model.

SMC.AuxSP *Semi-parametric mixture cure model with auxiliary subgroup survival information*

Description

Fit the semi-parametric mixture cure model with auxiliary subgroup survival probability information based on the control variate technique.

Usage

```
SMC.AuxSP(
  formula,
  cureform,
  sdata,
  aux = NULL,
  hetero = FALSE,
  N = Inf,
  latency = "PH",
  nboot = 400
)
```

Arguments

formula	a formula expression, of the form response ~ predictors. The response is a Surv object (from R package "survival") with right censoring. It is used to specify the covariate (risk factor) effects on the failure time of uncured subjects. See the documentation for survreg and Surv in R package survival for details. The expression to the right of the "~" specifies the effect of covariates on the failure time of uncured patients.
cureform	indicator function a formula expression, of the form cureform ~ predictors. It is used to specify the effects of covariates on the cure rate. Note that a covariate is allowed to be used in both formula and cureform.
sdata	a survival dataset (dataframe) in which to interpret the variables named in the formula and the cureform.
aux	indicates the historical aggregated statistics. It is a list of lists, and each sub-list represents auxiliary information from the same time point. We combine multiple time points together and each time point contains the following four elements tstar the time point that the auxiliary information was calculated at; sprob auxiliary subgroup survival rates for each subgroup at the current time point; gfunc a function used to identify the subgroup.
hetero	denotes a logical value. If it is TRUE, the penalization will be applied to identify the potential heterogeneous auxiliary subgroup survival rates and make a refinement to the final estimator.
N	records the sample size of the external sdata that we extract auxiliary information from. The default value is N = Inf, which is the case that the uncertainty is ignored. If N is not Inf, the method that takes the uncertainty into consideration will be adopted automatically.
latency	specifies the model used in latency part. It can be PH which represents the proportional hazards model, or AFT which represents the accelerated failure time model. The default is the PH mixture cure model, that is, latency = "PH"
nboot	specifies the number of bootstrap sampling. The default nboot = 400.

Details

This is a function used to fit the semiparametric mixture cure model with auxilary subgrop survival information. The method used here is the control variate technique. The test statistic for evaluating

the homogeneity assumption will also be calculated.

Value

An object of class `SMC.AuxSP` is returned. Specifically, it contains: `model`, a description of the model we fit; `suffix`, a character that indicates the status of auxiliary information; `coefficients`, estimated parameters. For more friendly presentation, we provide a function `Print.SMC.AuxSP()` that can examine this newly defined class.

Examples

```
#---------------------------------------------------------------------#
# illustration via simulated dataset (from PH mixture cure model) ####
#---------------------------------------------------------------------#

## library
library(survival)
library(CureAuxSP)

## generate both the internal dataset of interest and the external dataset

# - the internal dataset
set.seed(1)
sdata.internal <- sdata.SMC(n = 300)
head(sdata.internal)

# - the external dataset
set.seed(1)
sdata.external <- sdata.SMC(n = 10000)

## prepare the auxiliary information based on the external dataset

# - define two functions for subgroup splitting
gfunc.t1 <- function(X,Z=NULL){
  rbind((X[,1] <  0 & X[,2] == 0), (X[,1] >= 0 & X[,2] == 0),
        (X[,1] <  0 & X[,2] == 1), (X[,1] >= 0 & X[,2] == 1))}
gfunc.t2 <- function(X,Z=NULL){rbind((X[,2] == 0), (X[,2] == 1))}

# - calculate subgroup survival rates
sprob.t1 <- Probs.Sub(tstar = 1, sdata = sdata.external,
                      G = gfunc.t1(X = sdata.external[,-c(1,2)]))
sprob.t2 <- Probs.Sub(tstar = 2, sdata = sdata.external,
                      G = gfunc.t2(X = sdata.external[,-c(1,2)]))
cat("Information at t* = 1:", sprob.t1, "\nInformation at t* = 2:", sprob.t2)

# - prepare the set that collects information about auxiliary data
aux <- list(
  time1 = list(tstar = 1, gfunc = gfunc.t1, sprob = c(0.73,0.70,0.88,0.83)),
  time2 = list(tstar = 2, gfunc = gfunc.t2, sprob = c(0.62,0.76)-0.20)
)
```

```
## fit the model without auxiliary information
set.seed(1)
sol.PHMC <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ X1 + X2, cureform = ~ X1,
  sdata = sdata.internal, aux = NULL, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC)

## fit the model with auxiliary information

# - ignore heterogeneity
set.seed(1)
sol.PHMC.Homo <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ X1 + X2, cureform = ~ X1,
  sdata = sdata.internal, aux = aux, hetero = FALSE, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC.Homo)

# - consider heterogeneity
set.seed(1)
sol.PHMC.Hetero <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ X1 + X2, cureform = ~ X1,
  sdata = sdata.internal, aux = aux, hetero = TRUE, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC.Hetero)



#-----------------------------------------------------------------------#
# illustration via real breast cancer dataset (from TCGA program) ####
#   - the R package "TCGAbiolinks" should be downloaded in advance
#   - see "10.18129/B9.bioc.TCGAbiolinks" for more help
#-----------------------------------------------------------------------#

## library
library(survival)
library(CureAuxSP)

## prepare the breast cancer dataset

# - download clinical data from the TCGA website
query <- TCGAbiolinks::GDCquery(project = "TCGA-BRCA",
                                data.category = "Clinical", file.type = "xml")
TCGAbiolinks::GDCdownload(query)
clinical <- TCGAbiolinks::GDCprepare_clinic(query, clinical.info = "patient")

# - a preparation
sdata.pre <- data.frame(
  yobs   = ifelse(!is.na(clinical[,'days_to_death']),clinical[,'days_to_death'],
                  clinical[,'days_to_last_followup'])/365,
  delta  = ifelse(!is.na(clinical[,'days_to_death']),1,0),
  ER     = ifelse(
```

```
    clinical[,'breast_carcinoma_estrogen_receptor_status']=='Positive',1,
    ifelse(clinical[,'breast_carcinoma_estrogen_receptor_status']=='Negative',0,NA)),
  Age    = clinical[,'age_at_initial_pathologic_diagnosis'],
  Race   = ifelse(clinical[,'race_list']=='BLACK OR AFRICAN AMERICAN','black',
                  ifelse(clinical[,'race_list']=='WHITE','white','other')),
  Gender = ifelse(clinical[,'gender']=='FEMALE','Female',
                  ifelse(clinical[,'gender']=='MALE','Male',NA)),
  Stage  = sapply(
    clinical[,'stage_event_pathologic_stage'],
    function(x,pattern='Stage X|Stage IV|Stage [I]*'){
      ifelse(grepl(pattern,x),regmatches(x,regexpr(pattern,x)),NA)},USE.NAMES = FALSE)
)

# - extract covariates and remove undesiable subjects and NA
sdata.TCGA <- na.omit(
  sdata.pre[
    sdata.pre[,'yobs'] > 0
    & sdata.pre[,'Age'] <= 75
    & sdata.pre[,'Gender'] == "Female"
    & sdata.pre[,'Race'] %in% c('white')
    & sdata.pre[,'Stage'] %in% c('Stage I','Stage II','Stage III'),
    c('yobs','delta','Age','ER'),
  ]
)
rownames(sdata.TCGA) <- NULL

# - summary statistics of the internal dataset
summary(sdata.TCGA)

## plot a figure to show the existence of a cure fraction
# pdf("Figure_KM_TCGA_BRCA.pdf",width=8.88,height=6.66); {
plot(
  survival::survfit(survival::Surv(yobs, delta) ~ 1, data = sdata.TCGA),
  conf.int = T, mark.time = TRUE, lwd = 2,
  ylab = "Survival Probability", xlab = "Survival Time (in Years)",
  xlim = c(0,25), ylim = c(0,1)
)
# }; dev.off()

## fit the model without auxiliary information

# - rescale the Age variable
Age.Min <- min(sdata.TCGA$Age); Age.Max <- max(sdata.TCGA$Age)
sdata.TCGA$Age <- (sdata.TCGA$Age-Age.Min)/(Age.Max-Age.Min)

# - fit the model
set.seed(1)
sol.PHMC <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ Age + ER, cureform = ~ Age + ER,
  sdata = sdata.TCGA, aux = NULL, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC)
```

```
## fit the model with auxiliary information

# - prepare the auxiliary information
Age.Cut <- c(0,(c(40,50,60)-Age.Min)/(Age.Max-Age.Min),1)
gfunc.t1 <- function(X,Z){
  rbind((X[,1] >= Age.Cut[1] & X[,1] <  Age.Cut[2] & X[,2] == 1),
        (X[,1] >= Age.Cut[2] & X[,1] <  Age.Cut[3] & X[,2] == 1),
        (X[,1] >= Age.Cut[3] & X[,1] <  Age.Cut[4] & X[,2] == 1),
        (X[,1] >= Age.Cut[4] & X[,1] <= Age.Cut[5] & X[,2] == 1),
        (X[,1] >= Age.Cut[1] & X[,1] <  Age.Cut[2] & X[,2] == 0),
        (X[,1] >= Age.Cut[2] & X[,1] <  Age.Cut[3] & X[,2] == 0),
        (X[,1] >= Age.Cut[3] & X[,1] <  Age.Cut[4] & X[,2] == 0),
        (X[,1] >= Age.Cut[4] & X[,1] <= Age.Cut[5] & X[,2] == 0))}
gfunc.t2 <- function(X,Z){rbind((X[,2] == 1), (X[,2] == 0))}
aux <- list(
  time1 = list(tstar = 5, gfunc = gfunc.t1,
               sprob = c(0.810,0.935,0.925,0.950,0.695,0.780,0.830,0.850)),
  time2 = list(tstar = 10, gfunc = gfunc.t2, sprob = c(0.825,0.705))
)

# - ignore heterogeneity
set.seed(1)
sol.PHMC.Homo <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ Age + ER, cureform = ~ Age + ER,
  sdata = sdata.TCGA, aux = aux, hetero = FALSE, N = 1910, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC.Homo)

# - consider heterogeneity
set.seed(1)
sol.PHMC.Hetero <-  SMC.AuxSP(
  formula = Surv(yobs,delta) ~ Age + ER, cureform = ~ Age + ER,
  sdata = sdata.TCGA, aux = aux, hetero = TRUE, N = 1910, latency = "PH"
)
Print.SMC.AuxSP(object = sol.PHMC.Hetero)
```

Index