|  |  |
| --- | --- |
| PropJMvisDropCompMLE | R Documentation |

Joint modeling of longitudinal data, visiting process, and competing risks

**Description**

This function fits joint models of a normally distributed longitudinal marker, the visiting process, and competing risks under a shared parameter framework. The visiting process can be based either on a gap time or calendar time scale and the competing risks are modeled through cause-specific hazards. The fitting procedure is performed through maximum likelihood using Gauss–Hermite quadrature.

**Usage**

PropJMvisDropCompMLE(fitlme, fitCoxDrop1, fitCoxDrop2, fitCoxVis, maxit = 50, epsilon = 1e-06, nknotsDrop1 = 3, nknotsDrop2 = 3, nknotsVis = 3, nGH = 10, alpha = 0.05, timeVar = "times", Gauleg\_points = 30, useGauleg = F, epsDerXZ = 1e-07, ttBas = 0)

**Arguments**

|  |  |
| --- | --- |
| fitlme | an object from lme regarding the marker model (see also **Note**). |
| fitCoxCause1 | an object from coxph fitting proportional cause-specific hazards for cause 1. In the call to coxph(), you must specify the argument cluster(id) to define the rows of the same individual. See **Examples**. |
| fitCoxCause2 | an object from coxph fitting proportional cause-specific hazards for cause 2. In the call to coxph(), you must specify the argument cluster(id) to define the rows of the same individual. See **Examples**. |
| fitCoxVis | an object from coxph fitting the visiting process. A gap-time or calendar time scale is allowed (See **Details**). In the call to coxph(), you must specify the argument cluster(id) to define the rows of the same individual. See **Examples**. |
| maxit | Maximum number of BFGS iterations |
| epsilon | Tolerance for defining convergence, . |
| nknotsDrop1 | Number of internal knots of cubic B-spline basis for baseline cause-specific hazard function of cause 1 |
| nknotsDrop2 | Number of internal knots of cubic B-spline basis for baseline cause-specific hazard function of cause 2 |
| nknotsVis | Number of internal knots of cubic B-spline basis for baseline levels of the hazard function (gap time scale) or intensity function (calendar time scale) |
| nGH | Number of Gauss–Hermite points |
| alpha | Confidence level |
| timeVar | a character string indicating the time variable in the linear mixed model |
| useGauleg | logical, whether Gauss–Legendre quadrature is used to approximate the integrals with respect to time. If useGauleg=F, the 15-point Gauss-Kronrod rule is used. |
| Gauleg\_points | the number of Gauss–Legendre points. It is used if useGauleg=T, otherwise, it is ignored. |
| epsDerXZ | numeric scalar denoting the step size for the finite difference approximation in deriving . See **Details** |
| ttBas | time at which is evaluated. See **Details** |

**Details**

Function PropJMvisDropCompMLE can be used to jointly model a longitudinal marker, the visiting process, and competing risks. For the longitudinal responses, the linear mixed model represented by fitlme is assumed

where and denote the fixed-effect and random-effect design matrices, respectively, as specified in the fitlme object. Also, is interpreted as the “true” marker value at and .

The competing risk submodel is specified through the fitCoxCause1 and fitCoxCause2 objects. Specifically, the cause-specific hazard functions are assumed to be, for ,

where and are association parameters relating cause-specific hazard to and (the true marker value at baseline and current true marker slope), respectively. denotes the derivative of at . can be evaluated at different time points (see ttBas). Covariates and are derived from objects fitCoxCause1 and fitCoxCause2. Baseline cause-specific hazard functions are modeled by cubic B-spline of time (see nknotsCause1 and nknotsCause2). The integrals to calculate cumulative cause-specific hazard functions are approximated by either the Gauss–Legendre quadrature or the 15-point Gauss-Kronrod rule (see useGauleg and Gauleg\_points).

To specify a model for the visiting process based on gap times, the model fitCoxVis must have two arguments in the Surv function, e.g. Surv(gapTimesVis,deltaVis). Specifically, for the th gap time, a model of the form

is assumed, where and are association parameters relating the hazard of gap times to and , respectively. Covariates and are derived from objects fitCoxVis. The baseline hazard function for gap times is modeled by cubic B-splines of time (see nknotsVis). The integrals to calculate cumulative cause-specific hazard functions are approximated by either the Gauss–Legendre quadrature or the 15-point Gauss-Kronrod rule (see useGauleg and Gauleg\_points). When a gap-time model is adopted, timeVar must appear in the object fitCoxVis to define the analysis time (time since baseline) at the start of each interval.

To specify a model for the visiting process based on the intensity function using calendar time, the model fitCoxVis must have three arguments in the Surv function, e.g. Surv(tstart,tstop,deltaVis). Specifically, for , we assume a proportional intensity model of the form

The covariates and parameters have similar interpretations to those used in the gap-time model, but here denotes the baseline intensity function (see nknotsVis).

**Note**

1. The fitlme argument should represent a linear mixed model object with a general random-effects structure, i.e. only general positive definite structures for the covariance matrix of the random effects are allowed.

2. The fitlme object should not contain any within-group correlation structure (i.e., correlation argument of lme()) or within-group heteroscedasticity structure (i.e., weights argument of lme()).

3. It is assumed that the linear mixed-effects model fitlme and the Cox models fitCoxVis, fitCoxCause1, and fitCoxCause2 have been fitted to the same subjects. The data frames to which objects fitlme, fitCoxVis, fitCoxCause1, and fitCoxCause2 have been fitted should be sorted by the same numeric group variable (e.g. id=1,2,…). That is, it is assumed that the ordering of the subjects is the same for fitlme, fitCoxVis, fitCoxCause1, and fitCoxCause2, i.e., that the first rows of an individual in the data frame containing the event/visit times correspond to the first set of lines identified by the grouping variable in the data frame containing the repeated measurements, and so on.

### Examples

######################

### Fit the models ###

######################

# Import the function

source("PropJMvisDropCompMLE\_fit.R")

if (!require("nlme")) install.packages("nlme")

if (!require("survival")) install.packages("survival")

if (!require("matrixcalc")) install.packages("matrixcalc")

if (!require("splines2")) install.packages("splines2")

if (!require("MASS")) install.packages("MASS")

################################################################

### Fit the proposed model using a gap-time visiting process ###

################################################################

# Fit the linear mixed model

fitlme = try(lme(y ~ I(log(times+1)) + I((times/10)^3),

random = ~ I(log(times+1)) + I((times/10)^3)|id,data = dataLong,

control = list(apVar = T,opt = "optim" , returnObject = F,

maxIter = 100, msMaxIter = 100, niterEM = 150)),

silent = T)

summary(fitlme)

# Cox model for the gap times between visits

fitCoxVisGap <- coxph(Surv(gapTimesVis,deltaVis) ~ y + times + lag1\_gap + cluster(id),

data = dataLong, control = coxph.control(timefix = FALSE))

summary(fitCoxVisGap)

# Cox model for cause 1

fitCoxDrop1 <- coxph(Surv(tstart,tstop,deltaCause1) ~ y + group + cluster(id), data = dataLong,

control = coxph.control(timefix = FALSE))

summary(fitCoxDrop1)

# Cox model for cause 2

fitCoxDrop2 <- coxph(Surv(tstart,tstop,deltaCause2) ~ y + lag1\_gap + group + cluster(id), data = dataLong,

control = coxph.control(timefix = FALSE))

summary(fitCoxDrop2)

##############################################

### Fit the proposed model using gap times ###

##############################################

fitPropGap <- PropJMvisDropCompMLE(fitlme,fitCoxDrop1,fitCoxDrop2,fitCoxVisGap,

nknotsDrop1 = 1,nknotsDrop2 = 1,nknotsVis = 3,nGH = 4)

fitPropGap$sumLong

fitPropGap$sumVis

fitPropGap$sumDrop1

fitPropGap$sumDrop2

##################################################

### Fit the proposed model using calendar time ###

##################################################

fitCoxVisVcal <- coxph(Surv(tstart,tstop,deltaVis) ~ y + lag1\_gap + cluster(id),

data = dataLong,control = coxph.control(timefix = FALSE))

summary(fitCoxVisVcal)

# Fit the proposed model

fitPropCal <- PropJMvisDropCompMLE(fitlme, fitCoxDrop1,fitCoxDrop2, fitCoxVisVcal,

nknotsDrop1 = 1,nknotsDrop2 = 1,nknotsVis = 3,nGH = 4)

fitPropCal$sumLong

fitPropCal$sumVis

fitPropCal$sumDrop1

fitPropCal$sumDrop2