Package: miCoPTCM (via r-universe)

August 29, 2024

Type Package

Title Promotion Time Cure Model with Mis-Measured Covariates

Version 1.1

Date 2020-12-07

Author Aurelie Bertrand, Catherine Legrand, Ingrid Van Keilegom

Maintainer Aurelie Bertrand <aurelie.bertrand@uclouvain.be>

Imports MASS, nleqslv, survival, compiler, distr

Description Fits Semiparametric Promotion Time Cure Models, taking
into account (using a corrected score approach or the SIMEX
algorithm) or not the measurement error in the covariates,
using a backfitting approach to maximize the likelihood.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-07 14:40:02 UTC

Contents

---

miCoPTCM-package   *Promotion Time Cure Model with Mis-Measured Covariates*

Description

Fits Semiparametric Promotion Time Cure Models, taking into account (using a corrected score approach or the SIMEX algorithm) or not the measurement error in the covariates, using a backfitting approach to maximize the likelihood.

1

Details

Package:	miCoPTCM
Type:	Package
Title:	Promotion Time Cure Model with Mis-Measured Covariates
Version:	1.1
Date:	2020-12-07
Author:	Aurelie Bertrand, Catherine Legrand, Ingrid Van Keilegom
Maintainer:	Aurelie Bertrand <aurelie.bertrand@uclouvain.be>
Imports:	MASS, nleqslv, survival, compiler, distr
Description:	Fits Semiparametric Promotion Time Cure Models, taking into account (using a corrected score approach or the
License:	GPL-2

Index of help topics:

```
PTCMestimBF            Corrected score approach
PTCMestimSIMEX         SIMEX approach
miCoPTCM-package       Promotion Time Cure Model with Mis-Measured
                       Covariates
```

The survival model of interest is the promotion time cure model, i.e. a survival model which takes into account the existence of subjects who will never experience the event. The survival function of T, the survival time, is assumed to be improper:

$$S(t|\boldsymbol{x}) = P(T > t|\boldsymbol{X} = \boldsymbol{x}) = \exp\left\{-\theta(\boldsymbol{x})F(t)\right\},$$

where F is a proper baseline cumulative distribution function, θ is a link function with an intercept, here $\theta(\boldsymbol{x}) = \exp(\boldsymbol{x}^T\boldsymbol{\beta})$, and $\boldsymbol{x}$ is the vector of covariates. We work with the semiparametric version of this model, in which no known distribution is assumed for F. It can be shown that the nonparametric estimator of F is a step function which increases only at the failure times.

We assume that we have right censoring in our data, so that $Y = \min(T, C)$ is observed, where C is the censoring time.

The classical additive error model is assumed for the covariates, so that $\boldsymbol{W} = \boldsymbol{X} + \boldsymbol{U}$ is observed, where $\boldsymbol{W}$ is the vector of observed covariates and $\boldsymbol{U}$ is the vector of measurement errors. We assume that $\boldsymbol{U}$ is independent of $\boldsymbol{X}$ and $\boldsymbol{U}$ follows a continuous distribution with mean zero and known covariance matrix $\boldsymbol{V}$. It is also assumed that (T, C) and $\boldsymbol{W}$ are independent given $\boldsymbol{X}$.

Three possible estimation methods are available in this package. The corrected score approach of Ma and Yin (2008) is implemented in function PTCMestimBF. It consists in solving, through a backfitting procedure, the score equations in which the terms involving $\boldsymbol{x}$ are replaced by some terms involving $\boldsymbol{w}$ and $\boldsymbol{V}$. The naive method consists in not taking the measurement error in the covariates into account. The naive estimate is obtained by using function PTCMestimBF with a variance-covariance matrix of the error containing only zeros. Finally, the SIMEX algorithm applied to the promotion time cure model (Bertrand et al., 2015) is implemented in the function PTCMestimSIMEX. The SIMEX algorithm (Cook and Stefanski, 1994) is a generic and intuitive procedure allowing to estimate and reduce the bias in a model in which the covariates are measured with error. In this implementation, the naive estimator required by the procedure is the one of Ma and Yin (2008).

Author(s)

Aurelie Bertrand, Catherine Legrand, Ingrid Van Keilegom

Maintainer: Aurelie Bertrand <aurelie.bertrand@uclouvain.be>

References

Bertrand A., Legrand C., Carroll R.J., De Meester C., Van Keilegom I. (2015) Inference in a Survival Cure Model with Mismeasured Covariates using a SIMEX Approach. Submitted.

Cook J.R., Stefanski L.A. (1994) Simulation-Extrapolation Estimation in Parametric Measurement Error Models. Journal of the American Statistical Association, 89, 1314-1328. DOI: 10.2307/2290994

Ma, Y., Yin, G. (2008) Cure rate models with mismeasured covariates under transformation. Journal of the American Statistical Association, 103, 743-756. DOI: 10.1198/016214508000000319

PTCMestimBF *Corrected score approach*

Description

Fits a Semiparametric Promotion Time Cure Model, taking into account (using a corrected score approach) or not the measurement error in the covariates, using a backfitting approach to maximize the likelihood. Both methods were introduced in Ma and Yin (2008).

Usage

```
## Default S3 method:
PTCMestimBF(x, y, varCov, init, nBack=10000, eps=1e-8, multMaxTime=2,...)
## S3 method for class 'formula'
PTCMestimBF(formula, data=list(), ...)
## S3 method for class 'PTCMestimBF'
print(x,...)
## S3 method for class 'PTCMestimBF'
summary(object,...)
```

Arguments

x	a numerical matrix containing the explanatory variables as columns (without a column of 1s for the intercept).
y	the response, a survival object returned by the Surv function.
varCov	the square variance-covariance matrix of measurement error, with as many rows as regression parameters (including the intercept).
init	a numerical vector of initial values for the regression parameters.
nBack	an integer specifying the maximal number of iterations in the backfitting procedure.

eps	convergence criterion. Convergence is declared if the euclidian norm of the vector of changes in the estimated parameters and the euclidian norm of the score equations evaluated at these values are smaller than eps.
multMaxTime	a positive number controlling the time allowed, in one iteration of the backfitting procedure, to function nleqslv (used to solve the score equations) to converge.
formula	a formula object, in which the response is a survival object returned by the Surv function.
data	a dataframe containing the variables appearing in the model.
object	an object of class "PTCMestimBF", i.e., a fitted model.
...	not used.

Details

This method assumes normally distributed measurement error. The diagonal elements of the matrix varCov corresponding to covariates without error (as is the case for the intercept) have to be set to 0.

Value

An object of class PTCMestimBF, i.e. a list including the following elements:

coefficients	The estimated values of the regression parameters.
estimCDF	The estimated baseline cumulative distribution function.
vcov	The estimated variance-covariance matrix of the estimated regression parameters.
classObs	An integer vector of length 3: the number of censored individuals not considered as cured for the estimation, the number of events, and the number of individuals considered as cured for the estimation.
flag	Termination code: 1 if converged, 2 otherwise.
endK	Number of iterations performed in the backfitting procedure.

References

Bertrand A., Legrand C., Carroll R.J., De Meester C., Van Keilegom I. (2015) Inference in a Survival Cure Model with Mismeasured Covariates using a SIMEX Approach. Submitted.

Ma, Y., Yin, G. (2008) Cure rate models with mismeasured covariates under transformation. Journal of the American Statistical Association, 103, 743-756. DOI: 10.1198/016214508000000319

Examples

```
library("survival")
## Data generation
set.seed(123)
n <- 200
varCov <- matrix(nrow=3,ncol=3,0)
varCov[2,2] <- 0.1^1
X1 <- (runif(n)-.5)/sqrt(1/12)
```

```
V <- round(X1 + rnorm(n,rep(0,3),varCov[2,2]),7)# covariate with measurement error
Xc <- round(as.numeric(runif(n)<0.5),7) # covariate without measurement error

 # censoring times: truncated exponential distribution
C <- round(rexp(n,1/5),5)
Cbin <- (C>30)
while(sum(Cbin)>0)
{
 C[Cbin] <- round(rexp(sum(Cbin),1/5),5)
 Cbin <- (C>30)
}

expb <- exp(0.5+X1-0.5*Xc)
cure <- exp(-expb) # cure probabilities

 # event times with baseline cdf of a truncated exponential
U <- runif(n)
d <- rep(NA,n)
T <- round(-6*log( 1+ (1-exp(-20/6))*log(1-(1-cure)*U)/expb ),5)
T[(runif(n)<cure)] <- 99999 # cured subjects

Tobs <- rep(NA,n)
Tobs <- pmin(C,T) # observed times
Tmax <- max(Tobs[Tobs==T])
d <- (Tobs==T) # censoring indicator

Dat <- data.frame(Tobs,d,V,Xc)
#colnames(Dat) <- c("Tobs","d","V","Xc")


## Model estimation
fm <- formula(Surv(Tobs,d) ~ V + Xc)
resMY <- PTCMestimBF(fm, Dat, varCov=varCov, init=rnorm(3))
resMY
summary(resMY)
```

PTCMestimSIMEX   *SIMEX approach*

Description

Fits a Semiparametric Promotion Time Cure Model with mismeasured covariates, using the SIMEX algorithm based on a backfitting procedure. This approach is introduced in Bertrand et al. (2015).

Usage

```
## Default S3 method:
PTCMestimSIMEX(x, y, errorDistEstim=c("normal","student","chiSquare","laplace"),
paramDistEstim=NA, varCov=NA, nBack=10000, eps=1e-8, Nu=c(0,.5,1,1.5,2), B=50, init,
orderExtrap=2, multMaxTime=2,...)
```

```
## S3 method for class 'formula'
PTCMestimSIMEX(formula, data=list(),...)
## S3 method for class 'PTCMestimSIMEX'
print(x,...)
## S3 method for class 'PTCMestimSIMEX'
summary(object,...)
```

Arguments

x	a numerical matrix containing the explanatory variables as columns (without a column of 1s for the intercept).
y	the response, a survival object returned by the Surv function.
errorDistEstim	the distribution of the measurement error. See Details.
paramDistEstim	a scalar or a vector of length 2 containing the parameter(s) of the measurement error distribution, for non-Gaussian distributions. See Details.
varCov	the square variance-covariance matrix of measurement error, with as many rows as regression parameters (including the intercept), for Gaussian errors.
nBack	an integer specifying the maximal number of iterations in the backfitting procedure.
eps	convergence criterion.
Nu	a numerical vector containing the grid of lambda values, corresponding to the level of added noise.
B	the number of replications for each value in Nu.
init	a numerical vector of initial values for the regression parameters.
orderExtrap	a scalar or a numerical vector containing the degrees of the polynomials used in the extrapolation step.
multMaxTime	a positive number controlling the time allowed, in one iteration of the backfitting procedure, to function nleqslv (used to solve the score equations) to converge.
formula	a formula object, in which the response is a survival object returned by the Surv function.
data	a dataframe containing the variables appearing in the model.
object	an object of class "PTCMestimBF", i.e., a fitted model.
...	not used.

Details

More than one covariate can be subject to measurement error. However, in this implementation, all the errors must belong to the same family of distribution (specified with the argument errorDistEstim). Non-zero covariances are allowed between errors following a normal distribution. For the student, chi-squared and Laplace distributions, all variances are assumed to be equal (determined from paramDistEstim) and all covariances are assumed to be 0, even if the off-diagonal elements of vcov are not 0.

When using the laplace distribution, only one element in paramDistEstim is needed (if a vector of two elements is given, only the first element will be considered). With the student and

chiSquare distributions, two parameters are required, while none is required with the normal distribution. For the laplace distribution, the parameter is the inverse of the rate γ, where γ is such that $f(x) = \frac{1}{2}\gamma e^{-\gamma|x|}$. The first parameter of the Student distribution corresponds to the degrees of freedom, while the second parameter is a multiplicative factor such that the variance is $parameter_2^2 * \frac{parameter_1}{parameter_1-2}$. Similarly, for the chi-squared distribution, the first parameter gives the degrees of freedom, and the second one is a multiplicative factor yielding a variance of $2 \cdot param_2^2 \cdot param_1$.

Value

An object of class PTCMestimBF, i.e. a list including the following elements:

coefficients	The estimated values of the regression parameters.
var	The estimated variances of the estimated regression parameters.
classObs	An integer vector of length 3: the number of censored individuals not considered as cured for the estimation, the number of events, and the number of individuals considered as cured for the estimation.
estimNuBF	A matrix with as many rows as elements in Nu, containing, in each row, the average of the B estimates, for each regression parameter (columns).

References

Bertrand A., Legrand C., Carroll R.J., De Meester C., Van Keilegom I. (2015) Inference in a Survival Cure Model with Mismeasured Covariates using a SIMEX Approach. Submitted.

Cook J.R., Stefanski L.A. (1994) Simulation-Extrapolation Estimation in Parametric Measurement Error Models. Journal of the American Statistical Association, 89, 1314-1328. DOI: 10.2307/2290994

Ma, Y., Yin, G. (2008) Cure rate models with mismeasured covariates under transformation. Journal of the American Statistical Association, 103, 743-756. DOI: 10.1198/016214508000000319

Examples

```
## Not run:
library("survival")
## Data generation
set.seed(123)
n <- 200
varCov <- matrix(nrow=3,ncol=3,0)
varCov[2,2] <- 0.1^1
X1 <- (runif(n)-.5)/sqrt(1/12)
V <- round(X1 + rnorm(n,rep(0,3),varCov[2,2]),7)# covariate with measurement error
Xc <- round(as.numeric(runif(n)<0.5),7) # covariate without measurement error

 # censoring times: truncated exponential distribution
C <- round(rexp(n,1/5),5)
Cbin <- (C>30)
while(sum(Cbin)>0)
{
 C[Cbin] <- round(rexp(sum(Cbin),1/5),5)
 Cbin <- (C>30)
}
```

```
expb <- exp(0.5+X1-0.5*Xc)
cure <- exp(-expb) # cure probabilities

 # event times with baseline cdf of a truncated exponential
U <- runif(n)
d <- rep(NA,n)
T <- round(-6*log( 1+ (1-exp(-20/6))*log(1-(1-cure)*U)/expb ),5)
T[(runif(n)<cure)] <- 99999 # cured subjects

Tobs <- rep(NA,n)
Tobs <- pmin(C,T) # observed times
Tmax <- max(Tobs[Tobs==T])
d <- (Tobs==T) # censoring indicator

Dat <- data.frame(Tobs,d,V,Xc)

## Model estimation
fm <- formula(Surv(Tobs,d) ~ V + Xc)
resSimex <- PTCMestimSIMEX(fm, Dat, errorDistEstim="normal",
varCov=varCov,  nBack=10000, eps=1e-8,
Nu=c(0,.5,1,1.5,2), B=50, init=rnorm(3), orderExtrap=1:3, multMaxTime=2)
resSimex
summary(resSimex)
## End(Not run)
```

Index