Package: AFFECT (via r-universe)

August 31, 2024

Type Package

Title Accelerated Functional Failure Time Model with Error-Contaminated Survival Times

Version 0.1.2

Author Hsiao-Ting Huang <nikkihuang309700034@gmail.com> [cre,aut]
Li-Pang Chen <lchen723@nccu.edu.tw> [aut]

Maintainer Hsiao-Ting Huang <nikkihuang309700034@gmail.com>

Depends R (>= 3.3.1)

Imports stats,ggplot2

Description We aim to deal with data with measurement error in the response and misclassification censoring status under an AFT model. This package primarily contains three functions, which are used to generate artificial data, correction for error-prone data and estimate the functional covariates for an AFT model.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Date/Publication 2023-07-06 14:00:15 UTC

Repository https://meintraumus.r-universe.dev

RemoteUrl https://github.com/cran/AFFECT

RemoteRef HEAD

RemoteSha 37ecfaab8c3f458888242f613c5ea43d4bbca2e9

Contents

`AFFECT` *Accelerated Functional Failure Time Model with Error-Contaminated Survival Times*

Description

The package AFFECT, referred to Accelerated Functional Failure time model with Error-Contaminated survival Times, aims to recover the functional covariates under accelerated functional failure time models, where the data are subject to error-prone response and misclassified censoring status. This package primarily contains three functions. `data_gen` is applied to generate artificial data based on accelerated functional failure time models, including potential covariates, error-prone response and misclassified censoring status. `ME_correction` is used to do correction for error-prone response variable and misclassified censoring status, and `Boosting` is used to recover the functional covariates under accelerated functional failure time models.

Usage

```
AFFECT()
```

Details

This package aims to estimate functional covariates under an AFT models with error-prone response and and misclassified censoring status. The strategy is to derive an unbiased estimating function by the Buckley-James estimator with measurement error in response and misclassification in censoring status being corrected. Finally. the functional covariates as well as informative covariates under an AFT models can be derived by the boosting procedure.

Value

No return value, called for side effects.

`Boosting` *Estimation of Functional Forms of Covaraites under AFT Models*

Description

The function aims to select informative covariates under the AFT model and estimate their corresponding functional forms with survival time. Specifically, the first step in this function is to derive an unbiased estimating function by the Buckley-James method with corrected survival times and censoring status. After that, a boosting algorithm with the cubic-spline method is implemented to an unbiased estimating function to detect informative covariates and estimate the functional forms of covariates iteratively.

Usage

```
Boosting(data, iter = 50)
```

Arguments

- `data` — A `c(n,p+2)` dimension of data, where `n` is sample size and `p` is the number of covariates. The first column is survival time and second column is censoring status, and the other columns are covariates.
- `iter` — The iteration times of the boosting procedure. The default value = 50 and the iteration will stop when the absolute value of increment of every estimated value is small than 0.01.

Value

covariates The first ten covariates that are selected in the iteration.

functional_forms The functional forms of the first ten covariates that are selected in the iteration.

predicted_failure_time The predicted failure time of every sample

survival_curve Predicted survival curve of the sample.

Examples

```
## generate data with misclassification = 0.9 with n = 50, p = 6
## and variance of noise term is 0.75. The y* is is related to the first
## covariate.

b <- matrix(0,ncol=6, nrow = 1)
b[1,1] <- 1
data <- data_gen(n=50, p=6, pi_01=0.9, pi_10 = 0.9, gamma0=1,
gamma1=b, e_var=0.75)

## Assume that covariates are independent and observed failure time is
## related to first covariate with weight equals 1. And the scalar
## in the classical additive measurement error model is 1 and
## Misclassifcation probability = 0.9.

matrixb <- diag(6)
gamma_0 <-  1
gamma_1 <- matrix(0,ncol=6, nrow =1)
gamma_1[1,1] <- 1
data1 <- ME_correction(pi_10=0.9,pi_01=0.9,gamma0 = gamma_0,
gamma1 = gamma_1,
cor_covar=matrixb, y=data[,1],
indicator=data[,2], covariate = data[,3:8])
data1 <- cbind(data1,data[,3:8])

## Data in boosting procedure with iteration times =2

result <- Boosting(data=data1, iter=2)
```

data_gen *Generation of Artificial Data*

Description

The function generates a set of artificial data, including covariates generated by uniform distribution with an interval [0.5, 0.5], survival time and censoring status with measurement error and misclassifications. In this function, users can specify different degrees of measurement error that links observed survival time with true survival time, and links observed censoring status with true censoring status. Moreover, the accelerated functional failure time model considered in function is given by T=f(X1)+f(X2)+f(X3)+f(X4)+error, where T is log failure time and f(X1)=4*x1^2+x1, f(X2)=sin(6*x2),f(X3)=cos(6*x3)-1 and f(X4)=4*x4^3+x4^2.

Usage

```
data_gen(n, p, pi_01, pi_10, gamma0, gamma1, e_var)
```

Arguments

n	Sample size.
p	The number of covariates.
pi_01	Misclassifcation probability is P(Observed Censoring Status = 0 | Actual Censoring Status = 1).
pi_10	Misclassifcation probability is P(Observed Censoring Status = 1 | Actual Censoring Status = 0).
gamma0	A scalar that links the observed survival time and true survival time in the classical additive measurement error model y*=y+gamma0+gamma1*X+v, where y* is observed survival time and y is true survival time, and x is covariates and v is noise term.
gamma1	A p-dimensional vector of parameters in the additive measurement error model y*=y+gamma0+gamma1*X+v, where y* is observed survival time and y is true survival time, x is covariates and v is noise term.
e_var	The variance of noise term v in the additive measurement error model y*=y+gamma0+gamma1*X+v, where v is assumed to follow a normal distribution.

Value

generated_data c(n,p+2) dimensional data frame. The first column is observed survival time and second column is observed censoring status, and the other columns are covariates.

Examples

```
## Set the relationship between observed survival time
## and true survival time equals y*= y+1+X1+v, where the variance is
## 0.75 with n=500 and p=50 and misclassification probability=0.9.
```

```
a <- matrix(0,ncol=50, nrow = 1); a[1,1] <- 1
data <- data_gen(n=500, p=50, pi_01=0.9, pi_10 = 0.9, gamma0=1,
gamma1=a, e_var=0.75)
```

ME_correction — *Correction of Measurement Error in Survival time and Censoring Status.*

Description

This function aims to correct for measurement error in survival time and misclassification in censoring status. The key strategy in the function ME_correction includes regression calibration for survival time under additive measurement error models and the unbiased conditional expectation approach for censoring status under misclassification models. With information of parameters in measurement error models implemented, this function will give outputs with corrected survival time and censoring status.

Usage

```
ME_correction(
  pi_10,
  pi_01,
  gamma0,
  gamma1,
  cor_covar,
  indicator,
  yast,
  covariate
)
```

Arguments

pi_10	Misclassifcation probability is P(Observed Censoring Status = 1 | Actual Censoring Status = 0).
pi_01	Misclassifcation probability is P(Observed Censoring Status = 0| Actual Censoring Status = 1).
gamma0	A scalar that links the observed survival time and true survival time in the classical additive measurement error model y*=y+gamma0+gamma1*X+v, where y* is observed survival time and y is true survival time, and x is covariates and v is noise term.
gamma1	A p-dimensional vector of parameters in the additive measurement error model y*=y+gamma0+gamma1*X+v, where y* is observed survival time and y is true survival time, x is covariates and v is noise term.
cor_covar	A c(p,p) covariance matrix of a p-dimensional vector of covariates.
indicator	A n-dimensional vector of misclassified censoring status, such as the second column generated by the function gen_data.

yast — A n-dimensional vector of error-prone survival time, such as the first column generated by the function gen_data.

covariate — A c(n,p) matrix of covariates.

Value

correction_data A c(n,2) data frame. This first column is the corrected survival time, and the second column is the corrected censoring indicator.

Examples

```
## generate data with misclassification = 0.9 with n = 500,
## p = 50 and variance of noise term is 0.75. The y* is related
## to the first covariate.

a <- matrix(0,ncol=50, nrow = 1);a[1,1] <- 1
data <- data_gen(n=500, p=50, pi_01 = 0.9, pi_10 = 0.9,
gamma0=1, gamma1=a, e_var=0.75)

## Assume that covariates are independent and
## observed survival time is related to first covariate with
## weight equals 1. And the scalar in the classical additive
## measurement error model is 1 and is classifcation probability = 0.9.

matrixa <- diag(50)
gamma_0 <-  1 ; gamma_1 <- matrix(0,ncol=50, nrow =1); gamma_1[1,1] <- 1
corrected_data1 <- ME_correction(pi_10=0.9,pi_01=0.9,gamma0 = gamma_0,
gamma1 = gamma_1,
cor_covar=matrixa, y=data[,1],
indicator=data[,2], covariate = data[,3:52])
```

Index