Package ‘pseudoCure’

May 9, 2026

Title A Pseudo-Observations Approach for Analyzing Survival Data with
a Cure Fraction

Version 1.0.0

Date 2025-02-05

Description A collection of easy-to-use tools for regression analysis of survival data with a cure fraction proposed in Su et al. (2022) <doi:10.1177/09622802221108579>. The modeling framework is based on the Cox proportional hazards mixture cure model and the bounded cumulative hazard (promotion time cure) model. The pseudo-observations approach is utilized to assess covariate effects and embedded in the variable selection procedure.

License GPL (>= 2)

Depends R (>= 4.2.0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

NeedsCompilation yes

Encoding UTF-8

Imports Rcpp, MASS, ggplot2, ggpubr, rlang

Author Sy Han (Steven) Chiou [aut, cre],
Chien-Lin Su [aut],
Feng-Chang Lin [aut]

Maintainer Sy Han (Steven) Chiou <`schiou@smu.edu`>

Repository CRAN

Date/Publication 2025-02-06 13:10:02 UTC

Contents

`pseudoCure-package` *pseudoCure: A pseudo-observations approach for analyzing survival data with a cure fraction*

Description

A collection of easy-to-use tools for regression analysis of survival data with a cure fraction. The modeling framework is based on the Cox proportional hazards mixture cure model and the bounded cumulative hazard model. The pseudo-observations approach is utilized to assess covariate effects and embedded in the variable selection procedure.

Author(s)

Maintainer: Sy Han (Steven) Chiou <`schiou@smu.edu`>

Authors:

- Chien-Lin Su <`marksu740824@gmail.com`>
- Feng-Chang Lin <`flin@bios.unc.edu`>

`geelm` *Generalized Estimating Equation with Gaussian family*

Description

Fits a generalized estimating equation (GEE) model with Gaussian family with different link functions. The `geelm` function also supports LASSO or SCAD regularization.

Usage

```
geelm(
  formula,
  data,
  subset,
  id,
  link = c("identity", "log", "cloglog", "logit"),
  corstr = c("independence", "exchangeable", "ar1"),
  lambda,
  exclude,
  penalty = c("lasso", "scad"),
  nfolds = 5,
```

```
    nlambda = 200,
    binit,
    tol = 1e-07,
    maxit = 100
  )
```

Arguments

`formula`	A formula object starting with ~ for the model formula.
`data`	An optional data frame that contains the covariates and response variables.
`subset`	An optional logical vector specifying a subset of observations to be used in the fitting process.
`id`	A vector which identifies the clusters. If not specified, each observation is treated as its own cluster.
`link`	A character string specifying the model link function. Available options are `"identity"`, `"log"`, `"cloglog"`, and `"logit"`.
`corstr`	A character string specifying the correlation structure. Available options are `"independence"`, `"exchangeable"`, and `"ar1"`.
`lambda`	An option for specifying the tuning parameter used in penalization. When this is unspecified or has a `NULL` value, penalization will not be applied and `pCure()` will uses all covariates specified in the formulas. Alternatively, this can be specified as a vector numeric vector of non-negative values or "auto" for auto selection.
`exclude`	A binary numerical vector specifying which variables to exclude in variable selection. The length of `exclude` must match with the number of covariates. A value of 1 means to exclude in the variable selection.
`penalty`	A character string specifying the penalty function. The available options are `"lasso"` and `"scad"`.
`nfolds`	An optional integer value specifying the number of folds. The default value is 5.
`nlambda`	An optional integer value specifying the number of tuning parameters to try if `lambda = "auto"`.
`binit`	A optional numerical vector for the initial value. A zero vector is used when not specified.
`tol`	A positive numerical value specifying the absolute error tolerance in root search. Default at `1e-7`.
`maxit`	A positive integer specifying the maximum number of iteration. Default at 100.

Value

An object of class `"geelm"` representing a linear model fit with GEE.

Examples

```
gendat <- function() {
  id <- gl(50, 4, 200)
  visit <- rep(1:4, 50)
  x1 <- rbinom(200, 1, 0.6)
  x2 <- runif(200, 0, 1)
  phi <- 1 + 2 * x1
  rhomat <- 0.667^outer(1:4, 1:4, function(x, y) abs(x - y))
  chol.u <- chol(rhomat)
  noise <- as.vector(sapply(1:50, function(x) chol.u %*% rnorm(4)))
  e <- sqrt(phi) * noise
  y <- 1 + 3 * x1 - 2 * x2 + e
  dat <- data.frame(y, id, visit, x1, x2)
  dat
}

set.seed(1); str(dat <- gendat())
geelm(y ~ x1 + x2, id = id, data = dat, corstr = "ar1")
```

`km` *Kaplan-Meier estimate*

Description

This function exclusively returns the Kaplan-Meier survival estimate and the corresponding time points. It does not provide standard errors or any additional outputs that are typically included with the `survfit()` function.

Usage

```
km(time, status)
```

Arguments

`time`	A numeric vector for the observed survival times.
`status`	A numeric vector for the event indicator; 0 indicates right-censoring and 1 indicates events.

Value

A data frame with the Kaplan-Meier survival estimates, containing:

`time`	Time points at which the survival probability is estimated.
`surv`	Estimated survival probability at each time point.

Examples

```
data(Teeth500)
km(Teeth500$time, Teeth500$event)
```

`mzTest` *Maller-Zhou test*

Description

Performs the Maller-Zhou test.

Usage

```
mzTest(time, status)
```

Arguments

`time`	A numeric vector for the observed survival times.
`status`	A numeric vector for the event indicator; 0 indicates right-censoring and 1 indicates events.

Value

A list containing the Maller-Zhou test results, including the test statistic, p-value, and the number of observed events.

Examples

```
data(Teeth500)
mzTest(Teeth500$time, Teeth500$event)
```

`pCure` *Cure Rate Model with pseudo-observation approach*

Description

Fits either a mixture cure model or a bounded cumulative hazard (promotion time) model with pseudo-observation approach.

Usage

```
pCure(
  formula1,
  formula2,
  time,
  status,
  data,
  subset,
  t0,
  model = c("mixture", "promotion"),
```

```
  nfolds = 5,
  lambda1 = NULL,
  exclude1 = NULL,
  penalty1 = c("lasso", "scad"),
  lambda2 = NULL,
  exclude2 = NULL,
  penalty2 = c("lasso", "scad"),
  control = list()
)
```

Arguments

`formula1`	A formula object starting with ~ for the model formula. This specifies the covariates in the incidence component and the long-term component under the mixture cure model and the bounded cumulative model, respectively.
`formula2`	A formula object starting with ~ for the model formula. This specifies the covariates in the latency component and the short-term component under the mixture cure model and the bounded cumulative model, respectively.
`time`	A numeric vector for the observed survival times.
`status`	A numeric vector for the event indicator; 0 indicates right-censoring and 1 indicates events.
`data`	An optional data frame that contains the covariates and response variables (`time` and `event`).
`subset`	An optional logical vector specifying a subset of observations to be used in the fitting process.
`t0`	A vector of times, where the pseudo-observations are constructed. When not specified, the default values are the 10, 20, ..., 90th percentiles of uncensored event times.
`model`	A character string specifying the underlying model. The available functional form are `"mixture"` and `"promotion"` correspond to the mixture cure model and the bounded cumulative model, respectively.
`nfolds`	An optional integer value specifying the number of folds. The default value is 5.
`lambda1, lambda2`	An option for specifying the tuning parameter used in penalization. When this is unspecified or has a `NULL` value, penalization will not be applied and `pCure()` will uses all covariates specified in the formulas. Alternatively, this can be specified as a vector numeric vector of non-negative values or "auto" for auto selection.
`exclude1, exclude2`	A character string specifying which variables to exclude from variable selection. Variables matching elements in this string will not be penalized during the variable selection process. in variable selection.
`penalty1, penalty2`	A character string specifying the penalty function. The available options are `"lasso"` and `"scad"`.
`control`	A list of control parameters. See detail.

Value

An object of class `"pCure"` representing a cure model fit.

References

Su, C.-L., Chiou, S., Lin, F.-C., and Platt, R. W. (2022) Analysis of survival data with cure fraction and variable selection: A pseudo-observations approach *Statistical Methods in Medical Research*, **31**(11): 2037–2053.

Examples

```
## Function to generate simulated data under the PHMC model
simMC <- function(n) {
  p <- 10
  a <- c(1, 0, -1, 0, 0, 0, 0, 0, 0, 0) # incidence coefs.
  b <- c(-1, 0, 1, 0, 0, 0, 0, 0, 0, 0) # latency coefs.
  X <- data.frame(x = matrix(runif(n * p), n))
  X$x.3 <- 1 * (X$x.3 > .5)
  X$x.4 <- 1 * (X$x.4 > .5)
  X[,5:10] <- apply(X[,5:10], 2, qnorm)
  time <- -3 * exp(-colSums(b * t(X))) * log(runif(n))
  cure.prob <- 1 / (1 + exp(-2 - colSums(a * t(X))))
  Y <- rbinom(n, 1, cure.prob)
  cen <- rexp(n, .02)
  dat <- NULL
  dat$Time <- pmin(time / Y, cen)
  dat$Status <- 1 * (dat$Time == time)
  data.frame(dat, X)
}

## Fix seed and generate data
set.seed(1); datMC <- simMC(200)

## Oracle model with an unpenalized PHMC model
summary(fit1 <- pCure(~ x.1 + x.3, ~ x.1 + x.3, Time, Status, datMC))


## Penalized PHMC model with tuning parameters selected by 10-fold cross validation
## User specifies the range of tuning parameters
summary(fit2 <- pCure(~ ., ~ ., Time, Status, datMC, lambda1 = 1:10 / 10, lambda2 = 1:10 / 10))

## Penalized PHMC model given tuning parameters
summary(update(fit2, lambda1 = 0.7, lambda2 = 0.4))
```

`pCure.control` *Package options for pseudoCure*

Description

This function provides the fitting options for the `pCure()` function.

Usage

```
pCure.control(
  binit1 = NULL,
  binit2 = NULL,
  corstr = c("independence", "exchangeable", "ar1"),
  nlambda1 = 100,
  nlambda2 = 100,
  tol = 1e-07,
  maxit = 100
)
```

Arguments

`binit1`	Initial value for the first component. A zero vector will be used if not specified.
`binit2`	Initial value for the second component A zero vector will be used if not specified.
`corstr`	A character string specifying the correlation structure. The following are permitted: `"independence"`, `"exchangeable"`, and `"ar1"`.
`nlambda1, nlambda2`	An integer value specifying the number of lambda. This is only evoked when `lambda1 = "auto"` or `lambda2 = "auto"`.
`tol`	A positive numerical value specifying the absolute error tolerance in GEE algorithms.
`maxit`	An integer value specifying the maximum number of iteration.

Value

A list with control parameters.

See Also

`pCure`

`plot.geelm` *Plot method for 'geelm' objects*

Description

Plot method for 'geelm' objects

Usage

```
## S3 method for class 'geelm'
plot(x, type = c("residuals", "cv", "trace"), ...)
```

Arguments

`x`	An object of class 'pCure', usually returned by the 'pCure()' function.
`type`	A character string specifying the type of plot to generate. Available options are "residuals," "cv," and "trace," which correspond to the pseudo-residual plot, cross-validation plot, and trace plot for different values of the tuning parameter, respectively.
`...`	Other arguments for future extension.

Value

A ggplot object representing the residual plot, cross-validation plot, or the trace plot for an object of class `"geelm"`. This can be further modified using `"ggplot2"` functions.

`plot.pCure` *Plot method for 'pCure' objects*

Description

Plot method for 'pCure' objects

Usage

```
## S3 method for class 'pCure'
plot(x, part = "both", type = c("residuals", "cv", "trace"), ...)
```

Arguments

`x`	An object of class 'pCure', usually returned by the 'pCure()' function.
`part`	A character string specifies which component of the cure model to plot. The default is "both", which plots both the incidence and latency components if a mixture cure model was fitted, or both the long- and short-term effects if a promotion time model was fitted.
`type`	A character string specifying the type of plot to generate. Available options are "residuals," "cv," and "trace," which correspond to the pseudo-residual plot, cross-validation plot, and trace plot for different values of the tuning parameter, respectively.
`...`	Other arguments for future extension.

Value

A ggplot object representing the residual plot, cross-validation plot, or the trace plot for an object of class `"pCure"`. This can be further modified using `"ggplot2"` functions.

`Teeth500` *Dental data for illustration*

Description

Data on the survival of teeth with many predictors

Usage

```
data(Teeth500)
```

Format

A data frame containing the following variables:

time tooth survival time subject to right censoring.

event Tooth loss status: 1 = lost, 0 = not lost.

molar Molar indicator; 1 = molar tooth, 0 = non-molar tooth.

mobil Mobility score, on a scale from 0 to 5.

bleed Bleeding on probing, expressed as a percentage.

plaque Plaque score, expressed as a percentage.

pocket Periodontal probing depth.

cal Clinical Attachment Level.

fgm Free Gingival Margin.

filled Number of filled surfaces.

decay_new New decayed surfaces.

decay_recur Recurrent decayed surfaces.

crown Crown indicator; 1 = tooth has a crown, 0 = no crown.

endo Endodontic therapy indicator; 1 = endo therapy performed, 0 = no endo therapy.

filled_tooth Filled tooth indicator; 1 = filled, 0 = not filled.

decayed_tooth Decayed tooth indicator; 1 = decayed, 0 = not decayed.

total_tooth Total number of teeth.

gender Gender; 1 = male, 0 = female

diabetes Diabetes indicator; 1 = diabetes, 0 = no diabetes.

tobacco_ever Tobacco use indicator; 1 = had tobacco use, 0 = never had tobacco use.

A data frame with 500 observations and 20 variables.

Details

The data is a subset of the original dataset included in the `MST` package under the name `Teeth`. This subset contains the time to the first tooth loss due to periodontal reasons.

References

Calhoun, Peter and Su, Xiaogang and Nunn, Martha and Fan, Juanjuan (2018) Constructing Multivariate Survival Trees: The MST Package for R. *Journal of Statistical Software*, **83**(12).

Index