Package: curephEM (via r-universe)

October 28, 2026

Title NPMLE for Logistic-Cox Cure-Rate Model

Version 0.3.2

Description Expectation-Maximization (EM) algorithm for point estimation and variance estimation to the nonparametric maximum likelihood estimator (NPMLE) for logistic-Cox cure-rate model with left truncation and right- censoring. See Hou, Chambers and Xu (2017) <doi:10.1007/s10985-017-9415-2>.

License GPL-2

Depends R (>= 3.5), survival, Matrix

Suggests knitr, rmarkdown, xfun, devtools, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author Evan Ren [aut], Jue Hou [cre] (Marquis)

Maintainer Jue Hou <hou00123@umn.edu>

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

Date/Publication 2025-10-15 14:30:07 UTC

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

RemoteRef HEAD

RemoteSha 011b561dfbec2db7ccd72f31f98fa9c665c20ed7

Contents

cureph 2
cureph.control 4
cureph.object 5
sim.cureph.data 5
summary.cureph 6
Surv.cure 7
survpred.cureph 8

Index **10**

---

cureph *NPMLE for Logistic-CoxPH Cure-Rate Model*

Description

Fit NPMLE for Logistic-CoxPH Cure-Rate Model

Usage

```
cureph (formula, formula2 , data,  subset, na.action, init, control,
                method = c("EM"), singular.ok = TRUE,
                 var = c("Louis"),...)
```

Arguments

formula	a formula object for cure logistic model,with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv.cure function.
formula2	a formula object for event time CoxPH model The Response can be suppressed.
data	a data.frame in which to interpret the variables named in the formula and formula2, or in the subset and the weights argument.
subset	expression indicating which subset of the rows of data should be used in the fit. All observations are included by default.
na.action	a missing-data filter function. This is applied to the model.frame after any subset argument has been used. Default is options()$na.action.
init	a list of length 3, initial values of the iteration. Its compoenents are: a vector of formula coefficients,a vector offormula2 coefficients and a step.fun step function of baseline cummulative hazard function. Default initial value is the estimate from naive separate glm and coxph models.
control	Object of class cureph.control specifying iteration limit and other control options. Default is cureph.control(...).
method	a character string specifying the method for estimation. The default method is the EM algorithm.
singular.ok	logical value indicating how to handle collinearity in the model matrix. If TRUE, the program will automatically skip over columns of the X matrix that are linear combinations of earlier columns. In this case the coefficients for such columns will be NA, and the variance matrix will contain zeros. For ancillary calculations, such as the linear predictor, the missing coefficients are treated as zeros.
var	a character string specifying the method for variance estimation. The default method is derived from the Louis formula.
...	other parameters passed to cureph.control.

Details

The function can handle left truncation.

"BFGS-Newton" method optimizes the discretized nonparametric likelihood directly. It has 3 steps. First, it roughly search for a local maximal by BFGS method with default setup. Second, it rescales the parameters proportional to the reciprocal of current gradient in another BFGS optimization. Finally, it searches for a solution to the score equations through a Newton's method with Armijo line search.

The "EM" method uses a "ghost copy" algorithm to accomodate left-truncation.

Both variance estimation methods are asymptotically correct, but they have undercoverage issue in case of a small event number. "Louis" is recommended among the two.

See *Hou et al. (2016)* for detail.

Author(s)

Jue (Marquis) Hou

References

Hou, J., Chambers, C. and Xu, R. (2016). "A nonparametric maximum likelihood approach for partially observed cured data with left truncation and right-censoring". To be submitted.

See Also

cureph.object, summary.cureph, survpred.cureph

Examples

```
# A simulated data set
data('sim.cureph.data')
# Or you may generate another one
sim.cureph.data = cureph.simgen()

# Check out the true coefficients and baseline survival
attributes(sim.cureph.data)

# Fit logistic-CoxPH cure-rate model
fit=cureph(Surv.cure(time,time2,event,origin=0,end=20)~Z1+Z2+Z3+Z4,data=sim.cureph.data)

# Check out the summary
summary(fit)

# Baseline cummulative hazard
basehaz.cureph(fit)
```

`cureph.control` *Ancillary arguments for controling cureph fits*

Description

Assign the ancillary arguments for controling cureph fits

Usage

```
cureph.control(n.data,eps = 1e-09, toler.chol = .Machine$double.eps^0.75,
                 iter.max = 1000, toler.inf = eps^(1/3),
                 line.search=0.5,init.step = 1/n.data)
```

Arguments

`n.data`	Sample size of the data
`eps`	Iteration continues until the relative change in the log partial likelihood is less than eps. Must be positive.
`toler.chol`	Tolerance for detection of singularity during a Cholesky decomposion of the variance matrix, i.e., for detecting a redundant predictor variable.
`iter.max`	Maximum number of iterations to attempt for convergence.
`toler.inf`	Tolerance criteria for the warning message about a possible infinite coefficient value.
`line.search`	The scale of shrinkage in each Armijo line search step.
`init.step`	The initial size of step in Newton's method.

Details

`line.search` and `init.step` only matter when `method = "BFGS-Newton"`.

Value

a list containing the values of each of the above constants

Author(s)

Jue (Marquis) Hou

See Also

`cureph`

cureph.object *Cure-Rate Logistic-CoxPH Regression Object*

Description

This class of objects is returned by the cureph class of functions to represent a fitted cure-rate logistic-CoxPH model. Objects of this class have methods for the functions summary and survpred.

Arguments

coefficients	the vector of coefficients. If the model is over-determined there will be missing values in the vector corresponding to the redundant columns in the model matrix.
var	the variance matrix of the coefficients. Rows and columns corresponding to any missing coefficients are set to zero.
loglik	the log-likelihood with the final values of the coefficients.
baseline	the step-function for baseline cummulative hazard.
wald.test	the Wald test of whether the final coefficients differ from the initial values.
iter	number of iterations used.
means	vector of column means of the X matrix. Subsequent survival curves are adjusted to this value. Factors are set to be the baseline level.
n	the number of observations used in the fit.
nevent	the number of events (usually deaths) used in the fit.
method	the computation method used.
na.action	the na.action attribute, if any, that was returned by the na.action routine.
...	The object will also contain the following: terms.logistic, terms.cox, assign.logistic, assign.cox, df, var.levels, formula.logistic, formula.cox, contrasts.logistic, contrasts.cox, call, data, Y, origin and end.

See Also

cureph, summary.cureph, survpred.cureph.

sim.cureph.data *Simulated Data for* cureph

Description

A simulated data for cureph example.

Usage

```
data("sim.cureph.data")

cureph.simgen()
```

Format

A data frame with 200 observations on the following 7 variables.

`time` entry time.

`time2` exit time.

`event` binary event indicator.

`Z1` numerical vector.

`Z2` factor with levels '0' and '1'.

`Z3` factor with levels 'A', 'B' and 'C'.

`Z4` factor with levels '0', '1' and '2', colinear with `Z2` and `Z3`.

`attr(,"true.coef")` the true coefficients.

`attr(,"true.surv0")` the true baseline survival function.

`summary.cureph` *Summarizing Logistic-CoxPH Cure-Rate Model Fit*

Description

These functions are all methods for class `cureph` or `summary.cureph` objects.

Usage

```
## S3 method for class 'cureph'
summary(object,combine = T,...)

## S3 method for class 'summary.cureph'
print(x, digits = max(3, getOption("digits") - 3),
      signif.stars = getOption("show.signif.stars"),...)
```

Arguments

`object`	an object of class "`cureph`", usually, a result of a call to `cureph`.
`x`	an object of class "`summary.glm`", usually, a result of a call to `summary.glm`.
`combine`	logical; if `TRUE`, a Wald test table of each variable appearing in both models is returned.
`digits`	the number of significant digits to use when printing.
`signif.stars`	logical. If TRUE, "significance stars" are printed for each coefficient.
`...`	Other parameters to match the generic S3 method.

See Also

`cureph`

`Surv.cure` *Survival Object with Cure Portion*

Description

Create a Survival Object with Cure Portion

Usage

```
Surv.cure(time, time2, event, type=c("right","counting"), origin = 0, end = Inf)

## S3 method for class 'Surv.cure'
print(x, digit = getOption("digits"),...)
```

Arguments

`time`	for right censored data, this is the follow up time. For left truncated data, the first argument is the truncation time.
`event`	The status indicator, normally `0=alive`, `1=dead`. Other choices are `TRUE/FALSE` (`TRUE` = `death`) or `1/2` (`2=death`). Although unusual, the event indicator can be omitted, in which case all subjects are assumed to have an event.
`time2`	ending time of the counting process data only. Intervals are assumed to be open on the left and closed on the right, `(start, end]`. Event indicates whether an event occurred at the end of the interval.
`type`	character string specifying the type of censoring. Possible values are `"right"` and `"counting"`.
`origin`	for counting process data, the hazard function origin. This option was intended to be used in conjunction with a model containing time dependent strata in order to align the subjects properly when they cross over from one strata to another, but it has rarely proven useful.
`end`	the upper bound for event time. Any subject survives beyond this time is considered as an observed cure.
`x`	a 'Surv.cure' object.
`digit`	number of decimal digits in the output.
`...`	Other parameters to match the generic S3 method.

See Also

`Surv`

Examples

```
# A simulated data set
data("sim.cureph.data")
attach(sim.cureph.data)

# Construct and print a Surv.cure object
Surv.cure(time,time2,event,origin=0,end=20)

# Detach the simulated data
detach(sim.cureph.data)
```

`survpred.cureph` *Prediction for a Logistic-CoxPH Cure-Rate Model*

Description

Compute Predictions and Survival Curves from a Logistic-CoxPH Cure-Rate Model

Usage

```
## S3 method for class 'cureph'
survpred(object, newdata , time, center = F)
basehaz.cureph(object)

## S3 method for class 'survpred.cureph'
plot(x, pooled = T, censor = x$censored,...)
```

Arguments

`object`	a `cureph` object.
`newdata`	a data frame with the same variable names as those that appear in the `cureph` formula. It is also valid to use a vector, if the data frame would consist of a single row. The curve(s) produced will be representative of a cohort whose covariates correspond to the values in `newdata`. Default is the data frame used in the original fit for prediction and the mean of the covariates for survival curve. See `cureph.object`.
`time`	a numeric vector; all the time points to be evaluated in the survival table in addition to the observed event times.
`center`	logical; if `TRUE`, the baseline survival is evaluated at the mean of the covariates. Otherwise, it is evaluated at zero.
`x`	a `survpred.cureph` object.
`pooled`	logical; if `TRUE`, the marginal survival curve(s) for the cure-rate model will be plotted. Otherwise, the conditional survival curve(s) in the Cox part will be plotted.

censor	a numeric vector of the observed censoring times. If not NULL, a cross will mark each censored point on the survival curve.
...	Other parameters to match the generic S3 method.

Details

The means of factor variables are not computed. Instead, the baseline level is used.

The NPMLE baseline survival is estimated in a Breslow-Nelson-Aalen fashion.

Value

logistic.linear.predict	the linear predictions for cure indicator logistic model.
logistic.linear.prob	the estimated probability for NOT being cured.
cox.linear.predict	the linear predictions for event time CoxPH model.
cox.cumhaz	the baseline cummulative hazard for event time CoxPH model.
surv.cox	the marginal survival table for the cure-rate model.
surv.cureph	the conditional survival table for event time CoxPH model.
...	The object will also contain the following: censored, origin and end.

Author(s)

Jue (Marquis) Hou

See Also

cureph

Examples

```
library(curephEM)

# A simulated data set
data("sim.cureph.data")

# Fit logistic-CoxPH cure-rate model
fit=cureph(Surv.cure(time,time2,event,origin=0,end=20)~Z1+Z2+Z3+Z4,data=sim.cureph.data)

# The survival-prediction object
mysurv = survpred(fit,center=TRUE)

oldpar = par()
par(mfrow=c(1,2))
plot(mysurv)
plot(mysurv,FALSE)
par(oldpar)
```

Index

basehaz.cureph *(*survpred.cureph*)*, 8

cureph, 2, *4–6*, *9*
cureph.control, *2*, 4
cureph.object, *3*, 5, *8*
cureph.simgen *(*sim.cureph.data*)*, 5

plot.survpred.cureph *(*survpred.cureph*)*, 8
print.summary.cureph *(*summary.cureph*)*, 6
print.Surv.cure *(*Surv.cure*)*, 7

sim.cureph.data, 5
summary.cureph, *3*, *5*, 6
Surv, *7*
Surv.cure, *2*, 7
survpred *(*survpred.cureph*)*, 8
survpred.cureph, *3*, *5*, 8