Package 'ALassoSurvIC'

October 16, 2019

Type Package

Title Adaptive Lasso for the Cox Regression with Interval Censored and Possibly Left Truncated Data

Version 0.1.0

Author Chenxi Li, Daewoo Pak and David Todem

Maintainer Daewoo Pak <heavyrain.pak@gmail.com>

Description Penalized variable selection tools for the Cox proportional hazards model with interval censored and possibly left truncated data. It performs variable selection via penalized nonparametric maximum likelihood estimation with an adaptive lasso penalty. The optimal thresholding parameter can be searched by the package based on the profile Bayesian information criterion (BIC). The asymptotic validity of the methodology is established in Li et al. (2019 <doi:10.1177/0962280219856238>). The unpenalized nonparametric maximum likelihood estimation for interval censored and possibly left truncated data is also available.

Depends R (>= 3.5.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports Rcpp, parallel

LinkingTo Rcpp

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-10-16 12:20:05 UTC

R topics documented:

`ALassoSurvIC-package` *Variable selection procedure with the adaptive lasso for interval censored and possibly left truncated data*

Description

This package provides penalized variable selection tools for the Cox proportional hazards model with interval censored and possibly left truncated data. The main function `alacoxIC` performs the variable selection via a penalized nonparametric maximum likelihood estimation (PNPMLE) with an adaptive lasso penalty. The function also finds the optimal thresholding parameter automatically by minimizing the Bayesian information criterion (BIC). The unpenalized nonparametric maximum likelihood estimation for interval censored and possibly left truncated data is also available with the `unpencoxIC` function. The asymptotic validity of the methodology is established in Li et al. (2019).

Details

Package:	ALassoSurvIC
Type:	Package
Version:	1.0.0
Date:	2019-8-28
License:	GPL (>= 3)

Author(s)

Chenxi Li, Daewoo Pak and David Todem

References

Li, C., Pak, D., & Todem, D. (2019). Adaptive lasso for the Cox regression with interval censored and possibly left truncated data. Statistical methods in medical research. https://doi.org/10.1177/0962280219856238

See Also

`alacoxIC`; `unpencoxIC`

alacoxIC *Performing variable selection with an adaptive lasso penalty for interval censored and possibly left truncated data*

Description

The alacoxIC function performs variable selection with an adaptive lasso penalty for interval censored and possibly left truncated data. It performs penalized nonparametric maximum likelihood estimation through a penalized EM algorithm by following Li et al. (2019). The function searches the optimal thresholding parameter automatically, based on BIC. The variable selection approach, implemented by the alacoxIC function, is proven to enjoy the desirable oracle property introduced by Fan & Li (2001). The full details are available in Li et al. (2019).

Usage

```
## Default S3 method:
alacoxIC(lowerIC, upperIC, X, trunc, theta,
  normalize.X = TRUE, cl = NULL, max.theta = 1000, tol = 0.001,
  niter = 1e+05, string.cen = Inf, string.missing = NA, ...)
```

Arguments

...	for S4 method only.
lowerIC	A numeric vector for the lower limit of the censoring interval.
upperIC	A numeric vector for the upper limit of the censoring interval.
X	A numeric matrix for the covariates that will be used for variable selection.
trunc	A numeric vector for left truncated times. If supplied, the function performs the variable selection for interval censored and left truncated data. If trunc is missing, the data will be considered as interval censored data.
theta	A numeric value for the thresholding parameter. If theta is missing, the function automatically determines the thresholding parameter using a grid search algorithm, based on the Bayesian information criterion (BIC). See details below.
normalize.X	A logical value: if normalize.X = TRUE, the covariate matrix X will be normalized before fitting models. Default is TRUE.
cl	A cluster object created by makeCluster in the parallel package. If NULL, no parallel computing is used by default. See details below.
max.theta	A numeric value for the maximum value that a thresholding parameter can take when searching the optimal one. The algorithm will look up an optimal tunning parameter below max.theta. See details below.
tol	A numeric value for the absolute iteration convergence tolerance.
niter	A numeric value for the maximum number of iterations.
string.cen	A string indicating right censoring for upperIC. Default is Inf.
string.missing	A string indicating missing value. Default is NA.

Details

The grid search algorithm is used to find the optimal thresholding parameter using a grid search algorithm, based on BIC. Specifically, the `alacoxIC` function first searches the smallest integer thresholding parameter which all coefficient estimates are zero beween 1 and `max.theta` and then creates one hundred grid points by following the rule of Simon et al. (2011, Section 2.3). The one minimizing BIC among the one hundred candidates is chosen as the optimal thresholding parameter in the adaptive lasso estimation.

The cluster object, created by `makeCluster` in the `parallel` package, can be supplied with the `cl` argument to reduce computation time via parallel computing. The parallel computing will be used when searching the optimal thresholding parameter and calculating the hessian matrix of the log profile likelihood. How to use the parallel computing is illustrated in one of the examples given below.

Use the `baseline` function and the `plot` function to extract and plot the estimate of the baseline cumulative hazard function, respectively, from the object returned by the `alacoxIC`. The `plot` function also provides the plot of the estimated baseline survival function. See the usages in the examples given below.

References

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, 96(456), 1348-1360

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. Journal of statistical software, 39(5), 1.

Li, C., Pak, D., & Todem, D. (2019). Adaptive lasso for the Cox regression with interval censored and possibly left truncated data. Statistical methods in medical research. https://doi.org/10.1177/0962280219856238

See Also

`unpencoxIC`

Examples

```
library(ALassoSurvIC)

### Variable selection for interval censored data
data(ex_IC) # the 'ex_IC' data having 100 subjects and 6 covariates
lowerIC <- ex_IC$lowerIC
upperIC <- ex_IC$upperIC
X <- ex_IC[, -c(1:2)]

## Performing the variable selection algorithm using a single core
system.time(result <- alacoxIC(lowerIC, upperIC, X))

## Use parallel computing to reduce the computation time
library(parallel)
cl <- makeCluster(2L)  # making the cluster object 'cl' with two CPU cores
cl <- makeCluster(detectCores()) # run this code to use all CPU cores

system.time(result <- alacoxIC(lowerIC, upperIC, X, cl = cl))
```

```
result           # main result
baseline(result) # obtaining the baseline cumulative hazard estimate
plot(result)     # plotting the baseline estimated cumulative hazard function by default
plot(result, what = "survival")  # plotting the estimated baseline survival function

### Variable selection for interval censored and left truncated data
## Try following codes with the 'ex_ICLT' data example
data(ex_ICLT) # the 'ex_ICLT' data having 100 subjects and 6 covariates
lowerIC <- ex_ICLT$lowerIC
upperIC <- ex_ICLT$upperIC
trunc <- ex_ICLT$trunc
X <- ex_ICLT[, -c(1:3)]
result2 <- alacoxIC(lowerIC, upperIC, X, trunc)
result2

baseline(result2)
plot(result2)
plot(result2, what = "survival")
```

`ALassoSurvIC-internal` *Internal functions for the ALassoSurvIC package*

Description

Internal functions for the ALassoSurvIC package.

`baseline` *Obtaining the nonparametric maximum likelihood estimate (NPMLE) for the baseline cumulative hazard function*

Description

Extracting the NPMLE for the baseline cumulative hazard function from the input object. The input object must be the objects returned by the `alacoxIC` function or the `unpencoxIC` function. The support set over which the cumulative hazard increases is the same as that of the nonparametric maximum likelihood estimator, characterized by Alioum and Commenges (1996). The full details are available in Li et al. (2019).

Usage

```
## Default S3 method:
baseline(object, ...)
```

Arguments

...	for S4 method only.
object	the object must be the object retruned by the alacoxIC function or the unpencoxIC function.

Details

The estimator for the baseline cumulative hazard function increases only on some support sets, so called maximal intersections, and the NPMLE is indifferent to how it increases on the support sets. The definition of maximal intersections and other details are available in Alioum and Commenges (1996) and Li et al. (2019).

Value

A list with components:

support	The maximal intersections with a finite upper endpoint.
lambda	The jump sizes over the support set.
cum.lambda	The NPMLE of the baseline cumulative hazard function.

References

Alioum, A. and Commenges, D. (1996). A proportional hazards model for arbitrarily censored and truncated data. Biometrics 52, 512-524.

Li, C., Pak, D., & Todem, D. (2019). Adaptive lasso for the Cox regression with interval censored and possibly left truncated data. Statistical methods in medical research. https://doi.org/10.1177/0962280219856238

See Also

alacoxIC; unpencoxIC

Examples

```
library(ALassoSurvIC)

### Display the hazard function for the interval censored data
data(ex_ICLT) # the 'virtual' data having 100 subjects and 6 covariates
lowerIC <- ex_ICLT$lowerIC
upperIC <- ex_ICLT$upperIC
trunc <- ex_ICLT$trunc
X <- ex_ICLT[, -c(1:3)]
result <- unpencoxIC(lowerIC, upperIC, X, trunc)
baseline(result)
```

ex_IC *Virtual data set for interval censored data*

Description

The data ex_IC is a virtual data set created to show how to utilize the package. ex_IC is interval censored data. See ex_ICLT for interval censored and left truncated data.

Usage

```
data(ex_IC)
```

Format

The data have the following columns:

lowerIC The lower limit of the censoring interval.

upperIC The upper limit of the censoring interval.

trunc The vector of left truncated points.

X1 - X6 The covariate vectors used for variable selection.

See Also

ex_ICLT

Examples

```
library(ALassoSurvIC)
data(ex_IC) # 100 subjects and 6 covariates
print(ex_IC)
```

ex_ICLT *Virtual data set for interval censored and left truncated data*

Description

The data ex_ICLT is a virtual data set created to show how to utilize the package. ex_ICLT is interval censored and left truncated data. See ex_IC for interval censored data.

Usage

```
data(ex_ICLT)
```

Format

The data have the following columns:

`lowerIC` The lower limit of the censoring interval.

`upperIC` The upper limit of the censoring interval.

`trunc` The vector of left truncated points.

`X1 - X6` The covariate vectors used for variable selection.

See Also

ex_IC

Examples

```
library(ALassoSurvIC)
data(ex_IC) # 100 subjects and 6 covariates
print(ex_IC)
```

---

plot.alacoxIC *Plot method for* alacoxIC *object*

Description

The plot method for alacoxIC object for plotting the estimated baseline culmulative function and the estimated baseline survival function.

Usage

```
  ## S3 method for class 'alacoxIC'
plot(x, what = "cum.hazard", xlim, ylim, xlab, ylab, axes = FALSE, ...)
```

Arguments

...	for S4 method only.
x	An object of class alacoxIC returned by the alacoxIC function.
what	A character string specifying which function will be plotted. Default is "cum.hazard", which plots the estimated baseline cumulative hazard function. Set to "survival" to plot the estimated baseline survival function.
xlim	A vector with two elements for the limits of follow-up time.
ylim	A vector with two elements for the limits of y-axis.
xlab	A label for the x axis.
ylab	A label for the y axis.
axes	A logical value drawing both axes. Default is FALSE.

Details

The x argument must be the object returned by the alacoxIC function. Note that plot provides the conditional survival function for left truncated data, which is analogous to the function (5) of Alioum and Commenges (1996). See the usages in the examples given below.

References

Alioum, A. and Commenges, D. (1996). A proportional hazards model for arbitrarily censored and truncated data. Biometrics 52, 512-524.

Examples

```
library(ALassoSurvIC)

data(ex_ICLT) # interval censored and left truncated data
lowerIC <- ex_ICLT$lowerIC
upperIC <- ex_ICLT$upperIC
trunc <- ex_ICLT$trunc
X <- ex_ICLT[, -c(1:3)]
result <- alacoxIC(lowerIC, upperIC, X, trunc, theta = 1.5)

plot(result)  # plotting the estimated baseline cumulative hazard function by default
plot(result, what = "survival")  # plotting the estimated baseline survival function
```

---

plot.unpencoxIC *Plot method for* unpencoxIC *object*

Description

Plot method for unpencoxIC object for plotting the estimated baseline culmulative function and the estimated baseline survival function.

Usage

```
## S3 method for class 'unpencoxIC'
plot(x, what = "cum.hazard", xlim, ylim, xlab, ylab, axes = FALSE, ...)
```

Arguments

...	for S4 method only.
x	An object of class unpencoxIC returned by the unpencoxIC function.
what	A character string specifying which function will be plotted. Default is "cum.hazard", which plots the estimated baseline cumulative hazard function. Set to "survival" to plot the estimated baseline survival function.
xlim	A vector with two elements for the limits of follow-up time.
ylim	A vector with two elements for the limits of y-axis.

xlab	A label for the x axis.
ylab	A label for the y axis.
axes	A logical value drawing both axes. Default is FALSE.

Details

The x argument must be the object returned by the unpencoxIC function. Note that plot provides the conditional survival function for left truncated data, which is analogous to the function (5) of Alioum and Commenges (1996). See the usages in the examples given below.

References

Alioum, A. and Commenges, D. (1996). A proportional hazards model for arbitrarily censored and truncated data. Biometrics 52, 512-524.

Examples

```
library(ALassoSurvIC)

data(ex_ICLT) # interval censored and left truncated data
lowerIC <- ex_ICLT$lowerIC
upperIC <- ex_ICLT$upperIC
trunc <- ex_ICLT$trunc
X <- ex_ICLT[, -c(1:3)]
result <- unpencoxIC(lowerIC, upperIC, X, trunc)

plot(result)  # plotting the estimated baseline cumulative hazard function by default
plot(result, what = "survival")  # plotting the estimated baseline survival function
```

---

unpencoxIC *Performing unpenalized nonparametric maximum likelihood estimation for interval censored and possibly left truncated data*

Description

The unpencoxIC function performs unpenalized nonparametric maximum likelihood estimation. The function provides unpenalized nonparametric maximum likelihood estimates, standard errors, and 95% confidence intervals. The full details are available in Li et al. (2019).

Usage

```
## Default S3 method:
unpencoxIC(lowerIC, upperIC, X, trunc = NULL, normalize.X = TRUE,
  cl = NULL, tol = 0.001, niter = 1e+05, string.cen = Inf, string.missing = NA, ...)
```

Arguments

...	for S4 method only.
lowerIC	A numeric vector for the lower limit of the censoring interval.
upperIC	A numeric vector for the upper limit of the censoring interval.
X	A numeric matrix for the covariates that will be used for variable selection.
trunc	A numeric vector for left truncated time. If supplied, the function performs the variable selection for interval censored and left truncated data. If trunc is missing, the data will be considered as interval censored data.
normalize.X	A logical value: if normalize.X = TRUE, the covariate matrix X will be normalized before fitting models. Default is TRUE.
cl	A cluster object created by makeCluster in the parallel package. If NULL, no parallel computing is used by default. See details below.
tol	A numeric value for the absolute iteration convergence tolerance.
niter	A numeric value for the maximum number of iterations.
string.cen	A string indicating right censoring for upperIC. Default is Inf.
string.missing	A string indicating missing value. Default is NA.

Details

The cluster object, created by makeCluster in the parallel package, can be supplied with the cl argument to reduce computation time via parallel computing. The parallel computing will be used when calculating the hessian matrix of the estimates. How to use the parallel computing is illustrated in one of the examples given below.

Use the baseline function and the plot function to extract and plot the estimate of the baseline cumulative hazard function, respectively, from the object returned by the unpencoxIC. The plot function also provides the plot for estimated baseline survival function. See the usages in the examples given below.

References

Li, C., Pak, D., & Todem, D. (2019). Adaptive lasso for the Cox regression with interval censored and possibly left truncated data. Statistical methods in medical research. https://doi.org/10.1177/0962280219856238

See Also

alacoxIC

Examples

```
library(ALassoSurvIC)

### Getting the unpenalized NPMLE for interval censored data
data(ex_IC)
lowerIC <- ex_IC$lowerIC
upperIC <- ex_IC$upperIC
X <- ex_IC[, -c(1:2)]
```

```
system.time(result <- unpencoxIC(lowerIC, upperIC, X))

result           # main result
baseline(result) # obtaining the baseline cumulative hazard estimate
plot(result)     # plotting the estimated baseline cumulative hazard function by default
plot(result, what = "survival")  # plotting the estimated baseline survival function

## Use the parallel computing to reduce computational times
library(parallel)
cl <- makeCluster(2L)  # making the cluster object 'cl' with two CPU cores
cl <- makeCluster(detectCores()) # run this code to use all CPU cores
system.time(result <- unpencoxIC(lowerIC, upperIC, X, cl = cl))

### Getting the unpenalized NPMLE for interval censored and left truncated data
## Try following codes with the 'ex_ICLT' data example
data(ex_ICLT)
lowerIC <- ex_ICLT$lowerIC
upperIC <- ex_ICLT$upperIC
trunc <- ex_ICLT$trunc
X <- ex_ICLT[, -c(1:3)]
result2 <- unpencoxIC(lowerIC, upperIC, X, trunc)
result2

baseline(result2)
plot(result2)
plot(result2, what = "survival")
```

Index