

Package ‘coxphf’

March 3, 2018

Version 1.13

Date 2018-03-03

Title Cox Regression with Firth's Penalized Likelihood

Depends R (>= 3.0.2)

Imports survival

Description

Implements Firth's penalized maximum likelihood bias reduction method for Cox regression which has been shown to provide a solution in case of monotone likelihood (nonconvergence of likelihood function).

The program fits profile penalized likelihood confidence intervals which were proved to outperform

Wald confidence intervals.

License GPL

URL [http:](http://cemsiiis.meduniwien.ac.at/kb/wf/software/statistische-software/fccoxphf/)

[//cemsiiis.meduniwien.ac.at/kb/wf/software/statistische-software/fccoxphf/](http://cemsiiis.meduniwien.ac.at/kb/wf/software/statistische-software/fccoxphf/)

LazyLoad yes

NeedsCompilation yes

Author Georg Heinze [aut, cre],
Meinhard Ploner [aut]

Maintainer Georg Heinze <georg.heinze@meduniwien.ac.at>

Repository CRAN

Date/Publication 2018-03-03 09:50:20 UTC

R topics documented:

breast	2
coxphf	2
coxphfplot	6
coxphftest	8

Index	11
--------------	-----------

breast

Breast Cancer Data Set

Description

Provides the breast cancer data set as used by Heinze & Schemper, 2001. The data sets contains information on 100 breast cancer patients, including: survival time, survival status, Tumor stage, Nodal status, Grading, Cathepsin-D tumorexpression

Usage

```
data(breast)
```

Format

A data frame with 100 observations on the following 6 variables.

T a numeric vector

N a numeric vector

G a numeric vector

CD a numeric vector

TIME a numeric vector

CENS a numeric vector

References

Heinze, G., and Schemper, M. 2001. A solution to the problem of monotone likelihood. *Biometrics* 57(1) pp. 114-119.

Examples

```
data(breast)
```

coxphf

Cox Regression with Firth's Penalized Likelihood

Description

Implements Firth's penalized maximum likelihood bias reduction method for Cox regression which has been shown to provide a solution in case of monotone likelihood (nonconvergence of likelihood function). The program fits profile penalized likelihood confidence intervals which were proved to outperform Wald confidence intervals.

Usage

```
coxphf(formula = attr(data, "formula"), data = sys.parent(),
       pl = TRUE, alpha = 0.05, maxit = 50, maxhs = 5,
       epsilon = 1e-06, gconv=0.0001, maxstep = 0.5, firth = TRUE, adapt=NULL,
       penalty=0.5)
```

Arguments

formula	a formula object, with the response on the left of the operator, and the model terms on the right. The response must be a survival object as returned by the 'Surv' function (see its documentation in the survival package).
data	a data.frame in which to interpret the variables named in the 'formula' argument.
pl	specifies if confidence intervals and tests should be based on the profile penalized log likelihood (pl=TRUE, the default) or on the Wald method (pl=FALSE).
alpha	the significance level ($1-\alpha$ = the confidence level), 0.05 as default.
maxit	maximum number of iterations (default value is 50)
maxhs	maximum number of step-halvings per iterations (default value is 5). The increments of the parameter vector in one Newton-Rhaphson iteration step are halved, unless the new likelihood is greater than the old one, maximally doing maxhs halvings.
epsilon	specifies the maximum allowed change in standardized parameter estimates to declare convergence. Default value is 1e-6.
gconv	specifies the maximum allowed absolute value of first derivative of likelihood to declare convergence. Default value is 0.0001.
maxstep	specifies the maximum change of (standardized) parameter values allowed in one iteration. Default value is 0.5.
firth	use of Firth's penalized maximum likelihood (firth=TRUE, default) or the standard maximum likelihood method (firth=FALSE) for fitting the Cox model.
adapt	optional: specifies a vector of 1s and 0s, where 0 means that the corresponding parameter is fixed at 0, while 1 enables parameter estimation for that parameter. The length of adapt must be equal to the number of parameters to be estimated.
penalty	strength of Firth-type penalty. Defaults to 0.5.

Details

The phenomenon of monotone likelihood in a sample causes parameter estimates of a Cox model to diverge, with infinite standard errors. Therefore, classical maximum likelihood analysis fails; the usual Wald confidence intervals cover the whole range of real numbers. Monotone likelihood appears if there is single covariate or a linear combination of covariates such that at each event time, out of all individuals being at risk at that time, the individual with the highest (or at each event time the individual with the lowest) value for that covariate or linear combination experiences the event. It was shown that analysis by Firth's penalized likelihood method, particularly in conjunction with the computation of profile likelihood confidence intervals and penalized likelihood ratio tests is superior to maximum likelihood analysis. It completely removes the convergence problem mentioned in the paragraph on CONVERGENCE of the description of the function coxph. The formula may

involve time-dependent effects or time-dependent covariates. The response may be given in counting process style, but it cannot be used for multivariate failure times, as the program has no option to fit a robust covariance matrix. The user is responsible for the independency of observations within each risk set, i.e., the same individual should not appear twice within the same risk set.

The package `coxphf` provides a comprehensive tool to facilitate the application of Firth's penalized likelihood method to Cox regression analysis. The core routines are written in Fortran 90, (and to our knowledge this is the first package written in Fortran 90). Some description of the problem of monotone likelihood and Firth's penalized likelihood method as a solution can be found the web page <http://cemsiiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/fccoxphf/>.

Version 1.13 now includes a convergence check and issues a warning in case of non-convergence. Profile likelihood confidence intervals or the estimation of the penalized likelihood ratio p -values can be vulnerable to non-convergence for numerical issues. In case of non-convergence problems, we suggest to first compare the output values `iter.ci` with the input parameter `maxit`. Then, set `maxstep` to a smaller value, e.g., 0.1 and increase the number of allowed iterations to e.g. 500. This setting may slow down convergence for some of the confidence limits, but proved robust also in extreme data sets.

Value

<code>coefficients</code>	the parameter estimates
<code>alpha</code>	the significance level = 1 - confidence level
<code>var</code>	the estimated covariance matrix
<code>df</code>	the degrees of freedom
<code>loglik</code>	the null and maximized (penalized) log likelihood
<code>method.ties</code>	the ties handling method
<code>iter</code>	the number of iterations needed to converge
<code>n</code>	the number of observations
<code>y</code>	the response
<code>formula</code>	the model formula
<code>means</code>	the means of the covariates
<code>linear.predictors</code>	the linear predictors
<code>method</code>	the estimation method (Standard ML or Penalized ML)
<code>method.ci</code>	the confidence interval estimation method (Profile Likelihood or Wald)
<code>ci.lower</code>	the lower confidence limits
<code>ci.upper</code>	the upper confidence limits
<code>prob</code>	the p-values
<code>call</code>	the function call
<code>iter.ci</code>	the numbers of iterations needed for profile likelihood confidence interval estimation, and for maximizing the restricted likelihood for p-value computation.

Note

There exists an earlier version of coxphf for S-Plus, which is not able to involve time-dependent effects or the counting-process representation of survival times.

Author(s)

Georg Heinze and Meinhard Ploner

References

- Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80:27–38.
- Heinze G and Schemper M (2001). A Solution to the Problem of Monotone Likelihood in Cox Regression. *Biometrics* 57(1):114–119.
- Heinze G (1999). Technical Report 10/1999: The application of Firth’s procedure to Cox and logistic regression. Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna, Vienna. <http://cemsiiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/fccoxphf/>
- Heinze G and Ploner M (2002). SAS and SPLUS programs to perform Cox regression without convergence problems. *Computer Methods and Programs in Biomedicine* 67:217-223

See Also

coxphfplot, coxphftest

Examples

```
# fixed covariate and monotone likelihood
time<-c(1,2,3)
cens<-c(1,1,1)
x<-c(1,1,0)
sim<-cbind(time,cens,x)
sim<-data.frame(sim)
coxphf(sim, formula=Surv(time,cens)~x) #convergence attained!
#coxph(sim, formula=Surv(time,cens)~x) #no convergence!

# time-dependent covariate
test2 <- data.frame(list(start=c(1, 2, 5, 2, 1, 7, 3, 4, 8, 8),
                        stop =c(2, 3, 6, 7, 8, 9, 9, 9,14,17),
                        event=c(1, 1, 1, 1, 1, 1, 1, 0, 0, 0),
                        x =c(1, 0, 0, 1, 0, 1, 1, 1, 0, 0) ))

summary( coxphf( formula=Surv(start, stop, event) ~ x, pl=FALSE, data=test2))

# time-dependent effect
# the coxphf function can handle interactions of a (fixed or time-dependent)
# covariate with time
# such that the hazard ratio can be expressed as a function of time

summary(coxphf(formula=Surv(start, stop, event)~x+x:log(stop), data=test2, pl=FALSE, firth=TRUE))
```

```

# note that coxph would treat x:log(stop) as a fixed covariate
# (computed before the iteration process)
# coxphf treats x:log(stop) as a time-dependent covariate which changes (
# for the same individual!) over time

# time-dependent effect with monotone likelihood

test3 <- data.frame(list(start=c(1, 2, 5, 2, 1, 7, 3, 4, 8, 8),
                        stop =c(2, 3, 6, 7, 8, 9, 9, 9,14,17),
                        event=c(1, 0, 0, 1, 0, 1, 1, 0, 0, 0),
                        x    =c(1, 0, 0, 1, 0, 1, 1, 1, 0, 0) ))

summary( coxphf( formula=Surv(start, stop, event) ~ x+x:log(stop), pl=FALSE, maxit=400, data=test3))

# no convergence if option "firth" is turned off:
# summary( coxphf(formula=Surv(start, stop, event) ~ x+x:log(stop), pl=F,
#                data=test3, firth=FALSE)

data(breast)
fit.breast<-coxphf(data=breast, Surv(TIME,CENS)~T+N+G+CD)
summary(fit.breast)

```

coxphfplot

Plot the Penalized Profile Likelihood Function

Description

Plots the penalized profile likelihood for a specified parameter.

Usage

```

coxphfplot(formula = attr(data, "formula"), data = sys.parent(),
           profile, pitch = 0.05, limits, alpha = 0.05,
           maxit = 50, maxhs = 5, epsilon = 1e-06, maxstep = 0.5,
           firth = TRUE, penalty=0.5, adapt=NULL, legend = "center", ...)

```

Arguments

formula	a formula object, with the response on the left of the operator, and the model terms on the right. The response must be a survival object as returned by the 'Surv' function.
data	a data.frame in which to interpret the variables named in the 'formula' argument.
profile	a righthand formula specifying the plotted parameter, interaction or general term, e.g. ~ A or ~ A : C.

<code>pitch</code>	distances between the interpolated points in standard errors of the parameter estimate, the default value is 0.05.
<code>limits</code>	the range of the x-axis in terms of standard errors from the parameter estimate. The default values are the extremes of both confidence intervals, Wald and PL, plus or minus half a standard error, respectively.
<code>alpha</code>	the significance level ($1-\alpha$ the confidence level, 0.05 as default).
<code>maxit</code>	maximum number of iterations (default value is 50)
<code>maxhs</code>	maximum number of step-halvings per iterations (default value is 5). The increments of the parameter vector in one Newton-Rhaphson iteration step are halved, unless the new likelihood is greater than the old one, maximally doing <code>maxhs</code> halvings.
<code>epsilon</code>	specifies the maximum allowed change in penalized log likelihood to declare convergence. Default value is 0.0001.
<code>maxstep</code>	specifies the maximum change of (standardized) parameter values allowed in one iteration. Default value is 2.5.
<code>firth</code>	use of Firth's penalized maximum likelihood (<code>firth=TRUE</code> , default) or the standard maximum likelihood method (<code>firth=FALSE</code>) for fitting the Cox model.
<code>adapt</code>	optional: specifies a vector of 1s and 0s, where 0 means that the corresponding parameter is fixed at 0, while 1 enables parameter estimation for that parameter. The length of <code>adapt</code> must be equal to the number of parameters to be estimated.
<code>penalty</code>	strength of Firth-type penalty. Defaults to 0.5.
<code>legend</code>	if <code>FALSE</code> , legends in the plot would be omitted (default is <code>TRUE</code>).
<code>...</code>	other parameters to legend

Details

This function plots the profile (penalized) log likelihood of the specified parameter. A symmetric shape of the profile (penalized) log likelihood (PPL) function allows use of Wald intervals, while an asymmetric shape demands profile (penalized) likelihood intervals (Heinze & Schemper (2001)).

Value

A matrix of dimension $m \times 3$, with $m = 1/\text{pitch} + 1$. With the default settings, $m = 101$. The column headers are:

<code>std</code>	the distance from the parameter estimate in standard errors
<code>x</code>	the parameter value
<code>log-likelihood</code>	the profile likelihood at <code>x</code>

Author(s)

Georg Heinze and Meinhard Ploner

References

- Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80:27–38.
- Heinze G and Schemper M (2001). A Solution to the Problem of Monotone Likelihood in Cox Regression. *Biometrics* 57(1):114–119.
- Heinze G (1999). Technical Report 10/1999: The application of Firth’s procedure to Cox and logistic regression. Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna, Vienna. <http://cemsiiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/fccoxphf/>
- Heinze G and Ploner M (2002). SAS and SPLUS programs to perform Cox regression without convergence problems. *Computer Methods and Programs in Biomedicine* 67:217-223

See Also

coxphf

Examples

```
time<-c(1,2,3)
cens<-c(1,1,1)
x<-c(1,1,0)
sim<-cbind(time,cens,x)
sim<-data.frame(sim)
profplot<-coxphfplot(sim, formula=Surv(time,cens)~x, profile=~x)
```

coxphftest

Penalized Likelihood Ratio Test in Cox Regression

Description

Performs a penalized likelihood ratio test for hypotheses within a Cox regression analysis using Firth’s penalized likelihood.

Usage

```
coxphftest(formula = attr(data, "formula"),
            data = sys.parent(), test = ~., values,
            maxit = 50, maxhs = 5, epsilon = 1e-06,
            maxstep = 0.5, firth = TRUE, adapt=NULL, penalty=0.5)
```

Arguments

formula	a formula object, with the response on the left of the operator, and the model terms on the right. The response must be a survival object as returned by the ‘Surv’ function.
data	a data.frame in which to interpret the variables named in the ‘formula’ argument.

test	righthand formula of parameters to test (e.g. $\sim B + D$). As default the null hypothesis that all parameters are 0 is tested.
values	null hypothesis values, default values are 0. For testing the hypothesis $H_0: B_1=1$ and $B_4=2$ and $B_5=0$, specify test= $\sim B_1 + B_4 + B_5$ and values=c(1, 2, 0).
maxit	maximum number of iterations (default value is 50)
maxhs	maximum number of step-halvings per iterations (default value is 5). The increments of the parameter vector in one Newton-Rhaphson iteration step are halved, unless the new likelihood is greater than the old one, maximally doing maxhs halvings.
epsilon	specifies the maximum allowed change in penalized log likelihood to declare convergence. Default value is 0.0001.
maxstep	specifies the maximum change of (standardized) parameter values allowed in one iteration. Default value is 2.5.
firth	use of Firth's penalized maximum likelihood (firth=TRUE, default) or the standard maximum likelihood method (firth=FALSE) for fitting the Cox model.
adapt	optional: specifies a vector of 1s and 0s, where 0 means that the corresponding parameter is fixed at 0, while 1 enables parameter estimation for that parameter. The length of adapt must be equal to the number of parameters to be estimated.
penalty	strength of Firth-type penalty. Defaults to 0.5.

Details

This function performs a penalized likelihood ratio test on some (or all) selected parameters. It can be used to test contrasts of parameters, or factors that are coded in dummy variables. The resulting object is of the class coxphfetest and includes the information printed by the proper print method.

Value

testcov	the names of the tested model terms
loglik	the restricted and unrestricted maximized (penalized) log likelihood
df	the number of degrees of freedom related to the test
prob	the p-value
call	the function call
method	the estimation method (penalized ML or ML)

Author(s)

Georg Heinze and Meinhard Ploner

References

- Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80:27–38.
 Heinze G and Schemper M (2001). A Solution to the Problem of Monotone Likelihood in Cox Regression. *Biometrics* 57(1):114–119.

Heinze G (1999). Technical Report 10/1999: The application of Firth's procedure to Cox and logistic regression. Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna, Vienna. <http://cemsiiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/fccoxphf/>

Heinze G and Ploner M (2002). SAS and SPLUS programs to perform Cox regression without convergence problems. *Computer Methods and Programs in Biomedicine* 67:217-223

See Also

coxphf, coxphfplot

Examples

```
testdata <- data.frame(list(start=c(1, 2, 5, 2, 1, 7, 3, 4, 8, 8),
  stop =c(2, 3, 6, 7, 8, 9, 9, 9,14,17),
  event=c(1, 1, 1, 1, 1, 1, 1, 0, 0, 0),
  x1    =c(1, 0, 0, 1, 0, 1, 1, 1, 0, 0),
  x2    =c(0, 1, 1, 1, 0, 0, 1, 0, 1, 0),
  x3    =c(1, 0, 1, 0, 1, 0, 1, 0, 1, 0)))

summary( coxphf( formula=Surv(start, stop, event) ~ x1+x2+x3, data=testdata))

# testing H0: x1=0, x2=0

coxphftest( formula=Surv(start, stop, event) ~ x1+x2+x3, test=~x1+x2, data=testdata)
```

Index

*Topic **datasets**

breast, [2](#)

*Topic **survival**

coxphf, [2](#)

coxphfplot, [6](#)

coxphftest, [8](#)

breast, [2](#)

coxphf, [2](#)

coxphfplot, [6](#)

coxphftest, [8](#)

Surv (coxphf), [2](#)