Package 'coxphf'

April 17, 2009

Type Package

Title Cox regression with Firth's penalized likelihood

Version 1.0-2

Date 2007-04-10

Author R by Meinhard Ploner, Fortran by Georg Heinze

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

Depends R (>= 2.3.1), survival

Description Cox regression with Firth's penalized likelihood

License GPL

Repository CRAN

Date/Publication 2007-04-11 08:21:18

R topics documented:

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, maxstep = 2.5,
       firth = TRUE)
```

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.
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-α = 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 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.

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; parameter estimates and standard errors diverge such that the usual Wald confidence intervals cover the whole range of real numbers. This phenomenon has been described in the paragraph on CONVERGENCE of found in the help files of the coxph function. 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 provides an ideal solution to the problem of montone likleihood (Heinze and Schemper, 2001) as parameter estimates from this method are guaranteed to be finite. Profile penalized likelihood confidence intervals and penalized likelihood ratio tests are superior to their Wald counterparts in terms of coverage probability, size and power.

The `coxphf` function is able to handle time-dependent effects or time-dependent covariates. Time-dependent effects are specified by defining interactions of covariates with functions of time in the model `formula`. Time-dependent covariates can be accounted for by the counting-process representation of survival times. Please note that the function 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 one of the first R packages 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://www.meduniwien.ac.at/msi/biometrie/programme/fc.

Value

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

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.

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

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, 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, firth=FALSE))

# 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), data=test3))

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

`coxphfplot` *Plot the penalized profile likelihood function*

Description

This function 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 = 2.5, firth = TRUE,
           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`.
`pitch`	distances between the interpolated points in standard errors of the parameter estimate, the default value is 0.05.
`limits`	the range of the x-axis in terms of standard errors from the parameter estimate. By default, the x-axis range is [min(Wald Lower CL, PL lower CL)-s/2, max(Wald Upper CL, PL Upper CL)+s/2], with s denoting the standard error.
`alpha`	the significance level (1-α 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 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.
legend	if FALSE, legends in the plot would be omitted (default is TRUE).
...	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 x 3, with m = floor(1 + r / pitch), where r denotes the range of the x-axis in standard errors. The column headers are:

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

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.

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

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

This function 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 = 2.5, firth = TRUE)
```

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. `~ 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 H0: B1=1 and B4=2 and B5=0, specify `test= ~ B1 + B4 + B5` 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.

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 coxphftest 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.

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

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)
```

`decomposeSurv` *Decompose A Survival Formula*

Description

Decompose a `Surv` formula which can have time-dependent effects and return the model matrix and the time function matrix.

Usage

```
decomposeSurv(formula, data, sort)
```

Arguments

`formula`	A formula with `Surv` on the left-handed side and an arbitrary part on the right-handed side, including possibly time-dependent effects, e.g. `Surv(time, event) ~ x1 * x2 + I(x3^2) + x1 * log(time)`.
`data`	The `data.frame` with the columns needed to extract the formula.
`sort`	If TRUE, the returned model matrizes are sorted by stoptime and event.

Details

This function is mainly a helper function for `coxphf`, `coxphftest` and `coxphfplot`.

Value

`NTDE`	The number of time dependent effects.
`fac`	The factor matrix, see the value of `terms`.
`resp`	N x 3 - response matrix, see `model.extract` with parameter `response`.
`mm1`	Model matrix without time effects.
`timedata`	Matrix with time functions as columns.
`timeind`	Indicator of time-dependent effect: the i-th column of `timedata` interacts with the timeind[i]-th column of `mm1`.
`covnames`	The names of the covariates.
`ind`	An indicator vector. Elements with FALSE mean that these `covnames` are not part of the `formula`, but are needed to span a coherent model frame.

Author(s)

Meinhard Ploner, Georg Heinze

References

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.

Ploner, M. and Heinze, G. (2001). Technical Report 1/2001: An SPLUS library to perform Cox regression without convergence problems. Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna, Vienna.

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.

Examples

```
## prepares matrices for a model with time-dependent effects
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) ))

decomposeSurv(Surv(start,stop,event) ~ x + x * log(stop), data=test3)
```

Index