Package: coxphf (via r-universe)

October 28, 2026

Version 1.13.4

Date 2023-08-09

Title Cox Regression with Firth's Penalized Likelihood

Depends R (>= 3.5.0)

Imports survival, generics, tibble

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), see Heinze and Schemper (2001) and Heinze and Dunkler (2008). The program fits profile penalized likelihood confidence intervals which were proved to outperform Wald confidence intervals.

License GPL

URL https://cemsiis.meduniwien.ac.at/kb/wf/software/statistische-software/fccoxphf/

RoxygenNote 7.2.2

LazyData true

BugReports https://github.com/georgheinze/coxphf/issues/

NeedsCompilation yes

Author Georg Heinze [aut, cre], Meinhard Ploner [aut], Lena Jiricka [aut], Gregor Steiner [aut]

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

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

Date/Publication 2023-08-11 13:30:43 UTC

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

RemoteRef HEAD

RemoteSha e03ee72be0dc4ee826b8a3c481aa3708f417c483

Contents

coxphf-package 2
augment.coxphf 3
breast 4
coxphf 4
coxphfplot 8
coxphftest 10
glance.coxphf 12
tidy.coxphf 13

Index **14**

`coxphf-package` *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.

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 `https://cemsiis.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 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.

Author(s)

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

See Also

Useful links:

- https://cemsiis.meduniwien.ac.at/kb/wf/software/statistische-software/fccoxphf/
- Report bugs at https://github.com/georgheinze/coxphf/issues/

`augment.coxphf` *Augment a coxphf object*

Description

Augment a coxphf object

Usage

```
augment.coxphf(x, data = x$y, ...)
```

Arguments

`x`	A 'coxphf' object.
`data`	the dataset used to fit the model.
`...`	Unused, included for generic consistency only.

Value

A tidy [tibble::tibble()] summarizing component-level information about the model

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

```
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.

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,
  data,
  pl = TRUE,
  alpha = 0.05,
  maxit = 50,
  maxhs = 5,
  epsilon = 1e-06,
  gconv = 1e-04,
  maxstep = 0.5,
  firth = TRUE,
  adapt = NULL,
  penalty = 0.5
)
```

Arguments

formula	a formula object, with the response on the left 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-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 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.

Value

The object returned is of the class `coxphf` and has the following attributes:

`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
`terms`	the terms object used
`iter.ci`	the numbers of iterations needed for profile likelihood confidence interval estimation, and for maximizing the restricted likelihood for p-value computation.

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.

See Also

[coxphfplot, coxphftest]

Examples

```
# fixed covariate and monotone likelihood
library(survival)
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,
  data,
  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.
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. The default values are the extremes of both confidence intervals, Wald and PL, plus or minus half a standard error, respectively.
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.
penalty	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.
adapt	strength of Firth-type penalty. Defaults to 0.5.
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 $m \times 3$, with $m = 1/\texttt{pitch} + 1$. With the default settings, $m = 101$. 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.

Examples

```
library(survival)
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,
  data,
  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. ~ 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 todeclare 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 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)

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.

Examples

```
library(survival)
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)



# How to test total effect of a variable with time-dependent effect

# NOT RUN (works)
#fitt<- coxphf( formula=Surv(start, stop, event) ~ x1+x2+x3*stop, data=testdata, pl=FALSE)

#test <- coxphf(formula=Surv(start, stop, event) ~ x1+x2+x3*stop, data=testdata, adapt=c(1,1,0,0))

# PLR p-value for x3 + x3:stop
#pchisq((fitt$loglik[2]-test$loglik[2])*2, 2, lower.tail=FALSE)

#NOT RUN (does not work)
#test <- coxphf(formula=Surv(start, stop, event) ~ x1+x2+x3*stop, data=testdata, test=~x3+stop:x3)
```

`glance.coxphf` *Glance at a coxphf object*

Description

Glance at a coxphf object

Usage

```
## S3 method for class 'coxphf'
glance(x, ...)
```

Arguments

x	A ‘coxphf‘ object.
...	Unused, included for generic consistency only.

Value

A tidy [tibble::tibble()] summarizing component-level information about the model

---

tidy.coxphf *Tidy a coxphf object*

Description

Tidy a coxphf object

Usage

```
tidy.coxphf(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

x	A ‘coxphf‘ object.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to display coefficient estimates on an exponential scale.
...	Unused, included for generic consistency only.

Value

A tidy [tibble::tibble()] summarizing component-level information about the model

import tibble

Index

∗ **datasets**
    breast, 4
∗ **survival**
    coxphf-package, 2
_PACKAGE *(*coxphf-package*)*, 2

augment.coxphf, 3

breast, 4

coxphf, 4
coxphf-package, 2
coxphfplot, 8
coxphftest, 10

glance.coxphf, 12

tidy.coxphf, 13