Package 'CoxRidge'

February 27, 2015

Type Package

Title Cox Models with Dynamic Ridge Penalties

Version 0.9.2

Date 2015-02-12

Author Aris Perperoglou <aperpe@essex.ac.uk>

Maintainer Aris Perperoglou <aperpe@essex.ac.uk>

Description A package for fitting Cox models with penalized ridge-type partial likelihood. The package includes functions for fitting simple Cox models with all covariates controlled by a ridge penalty. The weight of the penalty is optimised by using a REML type-algorithm. Models with time varying effects of the covariates can also be fitted. Some of the covariates may be allowed to be fixed and thus not controlled by the penalty. There are three different penalty functions, ridge, dynamic and weighted dynamic. Time varying effects can be fitted without the need of an expanded dataset.

Depends survival,splines

LazyLoad yes

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2015-02-27 12:09:03

R topics documented:

CoxRidge-package *Fit penalized Cox models with fixed and time varying effects of the covariates.*

Description

Fits penalized Cox models using ridge penalties and a REML-type algorithm for optimization. The methods can be applied also to non-proportional hazards models where some or all of the covariates can be modelled with time varying effects.

Details

Package:	CoxRidge
Type:	Package
Version:	0.9.1
Date:	2013-06-20
License:	GPL (>= 2)

Author(s)

Aris Perperoglou <aperpe@essex.ac.uk> Maintainer: Aris Perperoglou <aperpe@essex.ac.uk>

References

Perperoglou A.(2013)*Cox models with dynamic ridge penalties on time varying effects of the covariates*. Statistics in Medicine, to appear

comp.graph *Plot a* `cox.ridge` *or* `cox.dynamic.ridge` *object.*

Description

Plots time varying effects of covariates.

Usage

```
comp.graph(obj, alpha = 0.05, xlab = "time", ylab = "X effect",
all.terms = TRUE, variable)
```

Arguments

obj	A cox.ridge or cox.dynamic.ridge object.
alpha	The alpha level used for confidence bands.
xlab	a title for the x axis
ylab	a title for the y axis
all.terms	When TRUE all time-varying effects variables are plotted.
variable	when all.terms=FALSE you need to specify which variable to be plotted. Use numbers 1,2,3... for first, second, third etc.

Details

Confidence bands are computed using the delta method.

References

Perperoglou A, le Cessie S, van Houwelingen HC (2006). *Reduced-rank hazard regression for modelling non-proportional hazards*. Statistics in Medicine: 25, 2831-2845

See Also

plot

Examples

```
data(ova)
attach(ova)
X <- cbind(karn,diam,figo)
X <- apply(X,2,function(x){(x-mean(x))/sqrt(var(x))})
Ft <- cbind(rep(1,nrow(X)),bs(time))
fit <- Dynamic.Ridge(time,death,X,Ft=Ft,fun="simple")
comp.graph(fit,all.terms=FALSE,variable=1)
par(mfrow=c(3,1))
comp.graph(fit)
```

---

cox.ridge *Fit a Cox model with a ridge penalty on all covariates*

Description

Fits a simple Cox model with a ridge penalty on all coefficients. The penalty weight can be optimized using a REML-type likelihood method or be chosen by the user.

Usage

```
cox.ridge(formula, lambda = 1, lambdaFixed = FALSE, eps = 10e-6, data = sys.parent(),
iter.max = 200, mon = FALSE)
```

Arguments

formula	a formula object, 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' function.
lambdaFixed	when TRUE the function does not seek to optimize the penalty weight.
lambda	When lambdaFixed is FALSE lambda is a scalar giving the starting value for the weight of the penalty. When lambdaFixed is true lambda is the chosen weight of the penalty.
eps	a small value. The criterion of convergance.
data	an optional data frame containing the variables named in the formula.
iter.max	maximum number of iterations, default is 200.
mon	when true the function prints out the computed lambda weigh in each iteration.

Value

cox.ridge returns an object of class "cox.ridge" The function print.cox.ridge is used to obtain and print a summary of the results. An object of class "cox.ridge" is a list containing the following components:

call	function call.
coef	the vector of coefficients.
loglik	the penalized log-likelihood of the model.
time	a vector with failure/censoring times.
death	a vector of status indicator.
X	a matrix of covariates.
iter	number of iterations used to maximise likelihood at a fixed lambda.
inter.it	number of iterations used to find optimal lambda.'
lambda	optimal weight of the penalty.
Hat	the hat matrix at convergance.
hess	the Hessian matrix of second derivatives.

Note

The function at the current form cannot handle missing values. The user has to take prior action with missing values before using this function.

Author(s)

Aris Perperoglou

References

Perperoglou A.(2013)*Cox models with dynamic ridge penalties on time varying effects of the covariates*. Statistics in Medicine, to appear

See Also

coxph, Dynamic.Ridge

Examples

```
data(ova)
attach(ova)
X <- cbind(karn,diam,figo)
X <- apply(X,2,function(x){(x-mean(x))/sqrt(var(x))})#standardize covariates
fit <- cox.ridge(Surv(time,death)~X,lambda=1)
fit ##regression coefficients correspond to the standardized covariates
```

CoxRidge internal *Internal* CoxRidge *functions.*

Description

Internal CoxRidge functions.

Details

These are not to be called by the user

Dynamic.Ridge *Fit a Cox model with time dependent effects of the covariates and penalized likelihood.*

Description

Fits a Cox model on which some, or all, of the covariates are allowed to have time varying effects. The likelihood is penalized either using a simple ridge penalty on all time varying covariates ("simple") or a dynamic ridge penalty ("dynamic") which includes the used time functions in the penalty. There is also the option for a "weighted" ridge penalty based on the baseline hazard.

Usage

```
Dynamic.Ridge(time, death, X, R, Ft, lambda = 0, fun = c("dynamic", "weighted","simple"),
eps = 1e-06, iter.max = 200, theta, mon = FALSE, lambdaFixed = FALSE)
```

Arguments

time	a vector containing failure/censoring times.
death	a vector containing the status indicator.
X	a matrix of time varying covariates.
R	an optional matrix of time fixed covariates. When R is missing then all covariates are assumed to be time varying.
Ft	a matrix containing the time functions. The first column must be constant.
lambda	When lambdaFixed is FALSE lambda is a scalar giving the starting value for the weight of the penalty. When lambdaFixed is true lambda is the chosen weight of the penalty.
fun	"simple", "dynamic", or "weigthed": types of penalty.
eps	a small value. The criterion of convergance.
iter.max	maximum number of iterations, default is 200.
theta	an optional matrix of starting values for coefficients of time varying effects.
mon	when true the function prints out the computed lambda weigh in each iteration.
lambdaFixed	when TRUE the function does not seek to optimize the penalty weight.

Value

`Dynamic.Ridge` returns an object of class "cox.dynamic.ridge" The function `print.cox.dynamic.ridge` is used to obtain and print a summary of the results. An object of class "cox.dynamic.ridge" is a list containing some the following components:

call	function call.
theta	a matrix of coefficient for the covariates with time varying effects.
fixed.coef	the vector of fixed effects coefficients.
loglik	the penalized log-likelihood of the model.
time	a vector with failure/censoring times.
death	a vector of status indicator.
X	the matrix of time varying covariates.
R	the matrix of fixed covariates.
Ft	the matrix of time functions
iter	number of iterations used to maximise likelihood at a fixed lambda.
inter.it	number of iterations used to find optimal lambda.'
lambda	optimal weight of the penalty.
Hat	the hat matrix at convergance.
h2	the non-penalized Hessian matrix of second derivatives.

Note

The function at the current form cannot handle missing values. The user has to take prior action with missing values before using this function.

Author(s)

Aris Perperoglou

References

Perperoglou A.(2013)*Cox models with dynamic ridge penalties on time varying effects of the covariates*. Statistics in Medicine, to appear

See Also

coxph, cox.ridge

Examples

```
data(GBSG)
attach(GBSG)
X <- cbind(age,grade)
R <- cbind(tumsize,posnodal,prm,esm)
X <- apply(X,2,function(x){(x-mean(x))/sqrt(var(x))}) #standardize covariates
R <- apply(R,2,function(x){(x-mean(x))/sqrt(var(x))}) #standardize covariates
Ft <- cbind(rep(1,nrow(X)),bs(rfst))

# a model with all covariates as time varying, simple penalty
fit.dr <- Dynamic.Ridge(rfst,cens,cbind(X,R),Ft=Ft,lambda=100,fun="simple",lambdaFixed=TRUE)
fit.dr  #regression coefficients correspond to the standardized covariates

# a model with all covariates as time varying, weighted penalty
fit.wdr <- Dynamic.Ridge(rfst,cens,cbind(X,R),Ft=Ft,lambda=324,theta=fit.dr$theta,
fun="weighted",mon=TRUE)
fit.wdr #regression coefficients correspond to the standardized covariates

# a model with fixed and time varying covariates
fit.dr <- Dynamic.Ridge(rfst,cens,X,R,Ft,lambda=150,fun="simple",lambdaFixed=TRUE)
fit.dr
```

GBSG *German Breast Cancer Study Group.*

Description

A data frame containing the observations from the GBSG study.

Usage

```
data(GBSG)
```

Format

This data frame contains the observations of 686 women:

`id` patient id 1...686.

`htreat` hormonal therapy, a factor at two levels 0 (no) and 1 (yes).

`age` age of the patients in years.

`menostat` menopausal status, a factor at two levels 1 (premenopausal) and 2 (postmenopausal).

`tumsize` tumor size (in mm).

`grade` tumor grade.

`posnodal` number of positive nodes.

`prm` progesterone receptor (in fmol).

`esm` estrogen receptor (in fmol).

`rfst` rfst recurrence free survival time (in days).

`cens` censoring indicator (0 censored, 1 event).

Source

Beyerle, R.L.A. Neumann and H.F. Rauschecker for the German Breast Cancer Study Group (1994). Randomized 2x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. Journal of Clinical Oncology, 12, 2086-2093. W. Sauerbrei and P. Royston (1999). Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. Journal of the Royal Statistics Society Series A, Volume 162(1), 71-94.

References

`package(mfp)`

Examples

```
data(GBSG)
str(GBSG)
```

---

ova *Ovarian cancer data set*

Description

Survival times of 358 ovarian cancer patients with information on three covariates, karnofsky status (karn), tumor diameter (diam), figo stage (figo) and patients id.

Usage

```
data(ova)
```

Format

A data frame with 358 observations on the following 6 variables.

`time` Survival times in days.

`death` Status indicator, 0=censored, 1=death.

`karn` Karnofsky status at the start of the follow up.

`figo` Figo stage.

`diam` Timour diameter.

`x` Patient id.

Source

Verweij, P. J. M. and Van Houwelingen H. C. (1993) *Cross-validation in survival analysis*. Statistics in Medicine: 12, 2305-2314

Examples

```
data(ova)
str(ova)
```

`print.cox.dynamic.ridge` *Print a* `cox.dynamic.ridge` *object.*

Description

Information describing the fitted `cox.dynamic.ridge` object.

Usage

```
## S3 method for class 'cox.dynamic.ridge'
print(x,...)
```

Arguments

`x`	a `cox.dynamic.ridge` object.
`...`	optional arguments passed to print.default; see the documentation on that method function.

See Also

`Dynamic.Ridge`.

Examples

```
data(GBSG)
attach(GBSG)
X <- cbind(age,grade)
R <- cbind(tumsize,posnodal,prm,esm)
X <- apply(X,2,function(x){(x-mean(x))/sqrt(var(x))}) #standardize covariates
R <- apply(R,2,function(x){(x-mean(x))/sqrt(var(x))}) #standardize covariates
Ft <- cbind(rep(1,nrow(X)),bs(rfst))

# a model with all covariates as time varying, simple penalty
fit.dr <- Dynamic.Ridge(rfst,cens,cbind(X,R),Ft=Ft,lambda=10,fun="simple",lambdaFixed=TRUE)
fit.dr
```

---

print.cox.ridge *Print a* cox.ridge *object.*

Description

Information describing the fitted cox.ridge object.

Usage

```
## S3 method for class 'cox.ridge'
print(x,...)
```

Arguments

x	a cox.ridge object.
...	optional arguments passed to print.default; see the documentation on that method function.

See Also

cox.ridge.

Examples

```
data(ova)
attach(ova)
X <- cbind(karn,diam,figo)
X <- apply(X,2,function(x){(x-mean(x))/sqrt(var(x))})#standardize covariates
fit <- cox.ridge(Surv(time,death)~X,lambda=1,lambdaFixed=TRUE)
fit ##regression coefficients correspond to the standardized covariates
```

Index