

Package ‘uniCox’

April 19, 2009

Title Univariate shrinkage prediction in the Cox model

Version 1.0

Author Rob Tibshirani

Description Univariate shrinkage prediction for survival analysis using in the Cox model.. Especially useful for high-dimensional data, including microarray data.

Maintainer Rob Tibshirani <tibs@stanford.edu>

Depends survival

LazyLoad false

LazyData false

License GPL-2

URL <http://www-stat.stanford.edu/~tibs/uniCox>

Repository CRAN

Date/Publication 2009-04-15 18:13:53

R topics documented:

predict.uniCox	2
uniCox	3
uniCoxCV	4

Index	6
--------------	----------

predict.uniCox *Function to compute the linear predictor from a coxUniv fit*

Description

Function to compute the linear predictor from a coxUniv fit

Usage

```
predict.uniCox(object, x, ...)
```

Arguments

object	Object returned by uniCox
x	Feature matrix, n obs by p variables
...	Included for compatibility with generic predict function

Details

This function compute the linear predictor from a coxUniv fit for a set of test features

Value

A matrix of dimension (number rows of x) by (number of lambda values), representing the predictions x

Source

Tibshirani, R. Univariate shrinkage in the Cox model for high dimensional data (2009). <http://www-stat.stanford.edu/~tibs/ftp/cus.pdf> To appear SAGMB.

Examples

```
library(survival)
# generate some data
x=matrix(rnorm(200*1000),ncol=1000)
y=abs(rnorm(200))
x[y>median(y),1:50]=x[y>median(y),1:50]+3
status=sample(c(0,1),size=200,replace=TRUE)

xtest=matrix(rnorm(50*1000),ncol=1000)
ytest=abs(rnorm(50))
xtest[ytest>median(ytest),1:50]=xtest[ytest>median(ytest),1:50]+3

statustest=sample(c(0,1),size=50,replace=TRUE)

# fit model
a=uniCox(x,y,status)
```

```
# get predictions on a test set
yhat=predict.uniCox(a,xtest)

# fit survival model to predicted values for 7th val of lambda
coxph(Surv(ytest,statusetest)~yhat[,7])
```

uniCox	<i>Function to fit a high dimensional Cox survival model using Univariate Shrinkage</i>
--------	---

Description

Function to fit a high dimensional Cox survival model using Univariate Shrinkage

Usage

```
uniCox(x,y,status,lamlist=NULL,nlam=20,del.thres=.01,max.iter=5)
```

Arguments

x	Feature matrix, n obs by p variables
y	Vector of n survival times
status	Vector of n censoring indicators (1= died or event occurred,0=survived, or event was censored)
lamlist	Optional vector of lambda values for solution path
nlam	Number of lambda values to consider
del.thres	Convergence threshold
max.iter	Maximum number of iterations for each lambda

Details

This function builds a prediction model for survival data with high-dimensional covariates, using the Univariate Shrinkage method.

Value

A list with components

lamlist	Values of lambda used
beta	Coef estimates, number of features by number of lambda values
mx	Mean of feature columns
vx	Square root of Fisher information for each feature
s0	Exchangeability factor for denominator of score statistic
call	Call to this function

Source

Tibshirani, R. Univariate shrinkage in the Cox model for high dimensional data (2009). <http://www-stat.stanford.edu/~tibs/ftp/cus.pdf> To appear SAGMB.

Examples

```
library(survival)
# generate some data
x=matrix(rnorm(200*1000),ncol=1000)
y=abs(rnorm(200))
x[y>median(y),1:50]=x[y>median(y),1:50]+3
status=sample(c(0,1),size=200,replace=TRUE)

xtest=matrix(rnorm(50*1000),ncol=1000)
ytest=abs(rnorm(50))
xtest[ytest>median(ytest),1:50]=xtest[ytest>median(ytest),1:50]+3

statustest=sample(c(0,1),size=50,replace=TRUE)

# fit uniCox model
a=uniCox(x,y,status)

# look at results
print(a)

# do cross-validation to examine choice of lambda
aa=uniCoxCV(a,x,y,status)

# look at results
print(aa)

# get predictions on a test set
yhat=predict.uniCox(a,xtest)

# fit survival model to predicted values
coxph(Surv(ytest,statustest)~yhat[,7])
```

uniCoxCV

Function to cross-validate a high dimensional Cox survival model using Univariate Shrinkage

Description

Function to cross-validate a high dimensional Cox survival model using Univariate Shrinkage

Usage

```
uniCoxCV(fit,x,y,status,nfolds=5,folds=NULL)
```

Arguments

<code>fit</code>	object returned by call to <code>uniCox</code>
<code>x</code>	Feature matrix, n obs by p variables
<code>y</code>	Vector of n survival times
<code>status</code>	Vector of n censoring indicators (1= died or event occurred, 0=survived, or event was censored)
<code>nfolds</code>	Number of cross-validation folds
<code>folds</code>	Optional list of sample numbers defining folds

Details

This function does cross-validation for a prediction model for survival data with high-dimensional covariates, using the Univariate Shrinkage method.

Value

A list with components

<code>devcvm</code>	Average drop in CV deviance for each lambda value
<code>ncallcvm=ncallcvm</code>	Average number of features with non-zero wts in the CV, for each lambda value
<code>se.devcvm</code>	Standard error of average drop in CV deviance for each lambda value
<code>devcv</code>	Drop in CV deviance for each lambda value
<code>ncallcv</code>	Number of features with non-zero wts in the CV, for each lambda value
<code>folds</code>	Indices for CV folds
<code>call</code>	Call to this function

Source

Tibshirani, R. Univariate shrinkage in the Cox model for high dimensional data (2009). <http://www-stat.stanford.edu/~tibs/ftp/cus.pdf> To appear SAGMB.

Examples

```
library(survival)
# generate some data
x=matrix(rnorm(200*1000),ncol=1000)
y=abs(rnorm(200))
x[y>median(y),1:50]=x[y>median(y),1:50]+3
status=sample(c(0,1),size=200,replace=TRUE)

# fit uniCox model
a=uniCox(x,y,status)

# do cross-validation to examine choice of lambda
aa=uniCoxCV(a,x,y,status)
```

Index

*Topic **regression**

`predict.uniCox`, 1

`uniCox`, 3

`uniCoxCV`, 4

*Topic **survival**

`predict.uniCox`, 1

`uniCox`, 3

`uniCoxCV`, 4

`predict.uniCox`, 1

`uniCox`, 3

`uniCoxCV`, 4