

Package ‘SurvRank’

August 26, 2015

Title Rank Based Survival Modelling

Version 0.1

Author Michael Laimighofer [aut, cre, ctb]

Maintainer Michael Laimighofer <michael.laimighofer@helmholtz-muenchen.de>

Description Estimation of the prediction accuracy in a unified survival AUC approach. Model selection and prediction estimation based on a survival AUC. Stepwise model selection, based on several ranking approaches.

License GPL-2

Depends R (>= 3.0.0), survival(>= 2.37-6), foreach(>= 1.0.11)

Imports ipred(>= 0.9-3), sampling(>= 2.6), glmnet(>= 1.9-8), utils(>= 3.1.0), survAUC(>= 1.0-5), randomForestSRC(>= 1.5.4), ggplot2(>= 1.0.0), reshape(>= 0.8), gplots(>= 2.16.0), mboost(>= 2.3-0), parallel(>= 3.2.0), doParallel(>= 1.0.8), rpart(>= 4.1-8)

LazyData true

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-26 14:22:13

R topics documented:

CVrankSurv_fct	2
fin_surv_model_fct	4
fsSurvRankBoost	5
fsSurvRankConc	5
fsSurvRankCox	6
fsSurvRankGlmnet	6
fsSurvRankRandCox	7
fsSurvRankRf	7

fsSurvRankRpart	8
fsSurvRankWang	8
glmnetRank	9
plot_CVsurv	9
riskscore_fct	10
risk_newdat	11
weighting_fct	13

Index	14
--------------	-----------

CVrankSurv_fct	<i>Main function of SurvRank.</i>
----------------	-----------------------------------

Description

Main input function for SurvRank.

Usage

```
CVrankSurv_fct(data, t.times, cv.out, cv.in, fs.method = "lasso.rank",
  nr.var = 10, sd1 = 0.95, ncl = 1, weig.t = T, n1 = 0.1,
  c.time = 10, ...)
```

Arguments

data	input of data as a list in the format: list.name\$x data.frame of covariates. list.name\$y response as a survival object, derived from Surv .
t.times	number of times the cross-validation should be repeated
cv.out	number of folds in outer cross validation loop (for estimation of the predictive accuracy)
cv.in	number of folds in inner cross validation loop (for model selection on the training set)
fs.method	Defaults to "lasso.rank". Ranking method to be applied. One of c("lasso.rank", "conc.rank", "rf.rank", "boost.rank").
nr.var	Number of variables up to which stepwise selection should be carried out. Has to be smaller than n number of observations.
sd1	factor to which sparser solutions should be chosen. Not maximum Survival AUC in inner loop is used in stepwise selection, instead $\max(\text{survAUC}) * \text{sd1}$ leading to sparser solutions
ncl	Defaults to 1. Number of clusters for parallel execution.
weig.t	Defaults to TRUE. Should a weighting of features be performed.
n1	used in weighting function if weig.t=T. Find details in weighting_fct
c.time	as defined in package survAUC time; a positive number restricting the upper limit of the time range under consideration.
...	arguments that can be passed to underlying functions, not used now

Details

details to follow

Value

Output of the CVrankSurv_fct, basically a list containing the following elements

method	ranking method
accuracy\$ranking	full ranking of all model estimations
accuracy\$pred.in	averaged inner AUCs of stepwise selection
accuracy\$pred.out	predictions of testset
accuracy\$used.rank	only used features according to stepwise selection
accuracy\$used.rank1se	only used features according to stepwise selection with factor sd1
accuracy\$auc.out	matrix of dimension cv.out times t.times of survival AUCs.
accuracy\$auc.out1se	matrix of dimension cv.out times t.times of survival AUCs with factor sd1.
rank.mat	matrix of ranks per feature. If not selected, it is set to number of features.
out.mat	0/1 matrix for features selected
out.mat1se	0/1 matrix for features selected with factor sd1 application
top1se	unweighted toplist with factor sd1
toplist	unweighted toplist
weighted	weighted toplist with applied weighting function
rank	toplist of ranked features according to ranks

Examples

```
## Simulating a survival data set
N=100; p=10; n=4
x=data.frame(matrix(rnorm(N*p),nrow=N,p))
beta=rnorm(n)
mx=matrix(rnorm(N*n),N,n)
fx=mx[,seq(n)]%*%beta/3
hx=exp(fx)
ty=rexp(N,hx)
tcens=1-rbinom(n=N,prob=.3,size=1)
y=Surv(ty,tcens)
data=list()
data$x<-x; data$y<-y
## Ranking the features according to their significance in the univariate cox models
out.cox<-CVrankSurv_fct(data,2,3,3,fs.method="cox.rank")
```

```
## Ranking the features according to the LASSO algorithm
## Not run:
out.lasso<-CVrankSurv_fct(data,2,5,5,fs.method="lasso.rank")
## End(Not run)
```

fin_surv_model_fct *Building the final survRank model*

Description

This function builds the final survRank model

Usage

```
fin_surv_model_fct(f, data, cv.out, nr.var = 10, c.time = NA)
```

Arguments

f	ranking approach function. One of fsSurvRankConc, fsSurvRankGlmnet, fsSurvRankRf, fsSurvRankBoost, fsSurvRankCox, fsSurvRankRandCox, fsSurvRankRpart, fsSurvRankWang, fsSurvRankitBMA or NA, no calculation
data	same list used as input in CVrankSurv_fct
cv.out	number of folds in outer cross validation loop (for estimation of the predictive accuracy)
nr.var	Number of variables up to which stepwise selection should be carried out. Has to be smaller than n number of observations.
c.time	as defined in UnoCsurvAUC time; a positive number restricting the upper limit of the time range under consideration

Details

details to follow

Value

Output of the riskscore_fct, basically a list containing the following elements

ranking	ranking of the variables in the data set using the ranking approach function
used.rank	variables used in the survRank model according to the number of parameters to be used
model	cox regression model for the selected features
sum.model	summary of the fitted cox model

fsSurvRankBoost	<i>Boost ranking function</i>
-----------------	-------------------------------

Description

This function ranks the input features according to their selection probability in additive models via component-wise boosting

Usage

```
fsSurvRankBoost(x, y, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
...	other arguments, not used now

fsSurvRankConc	<i>Concordance ranking function</i>
----------------	-------------------------------------

Description

This function ranks the input features with the concordance measure.

Usage

```
fsSurvRankConc(x, y, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
...	other arguments, not used now

fsSurvRankCox	<i>Cox ranking function</i>
---------------	-----------------------------

Description

This function ranks the input features according to their significance in the univariate cox models

Usage

```
fsSurvRankCox(x, y, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
...	other arguments, not used now

fsSurvRankGlmnet	<i>L1 (lasso) ranking function</i>
------------------	------------------------------------

Description

This function ranks the input features with the lasso algorithm in glmnet.

Usage

```
fsSurvRankGlmnet(x, y, alp = 1, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
alp	alpha value in glmnet (elasticnet mixing parameter)
...	other arguments, not used now

fsSurvRankRandCox *Random Cox ranking function*

Description

This function ranks the input features according to their median significance in cox models with randomly selected nmax variables

Usage

```
fsSurvRankRandCox(x, y, bb = 500, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
bb	set to 500. Number of random subsamples
...	other arguments, not used now

fsSurvRankRf *Random forest ranking function*

Description

This function ranks the input features with a random forest according to the variable importance.

Usage

```
fsSurvRankRf(x, y, ...)
```

Arguments

x	x matrix or data.frame
y	response y as a survival object, generated with Surv()
...	other arguments, not used now

fsSurvRankRpart *Rpart ranking function*

Description

This function ranks the input features according to their importance in recursive partitioning and regression trees fitted with the function `rpart`

Usage

```
fsSurvRankRpart(x, y, ...)
```

Arguments

<code>x</code>	<code>x</code> matrix or <code>data.frame</code>
<code>y</code>	response <code>y</code> as a survival object, generated with <code>Surv()</code>
<code>...</code>	other arguments, not used now

fsSurvRankWang *Random Cox ranking function*

Description

This function ranks the input features according to their mean significance in univariate cox models for a randomly selected subset of the data

Usage

```
fsSurvRankWang(x, y, bb = 400, ns = 80, ...)
```

Arguments

<code>x</code>	<code>x</code> matrix or <code>data.frame</code>
<code>y</code>	response <code>y</code> as a survival object, generated with <code>Surv()</code>
<code>bb</code>	repeatelements of the bootstrap
<code>ns</code>	sample size per bootstrap
<code>...</code>	other arguments, not used now

glmnetRank	<i>Ranks features of a previously fitted glmnet object</i>
------------	--

Description

This function ranks the input features of a previously fitted glmnet object. Ranking according to the first occurrence in the lambda path or effect sizes at the end of the path.

Usage

```
glmnetRank(glmnet, first = T, names = T, ...)
```

Arguments

glmnet	previously fitted glmnet object
first	Defaults to TRUE. TRUE ranking based on occurrence. FALSE based on effect sizes
names	Defaults to TRUE. TRUE returns feature names. FALSE returns coefficients
...	other arguments, not used now

plot_CVsurv	<i>Main function of SurvRank</i>
-------------	----------------------------------

Description

This function creates a pdf file containing some summary plots of the survival ranking analysis

Usage

```
plot_CVsurv(cv.ob, data, file = "test.surv.pdf", ...)
```

Arguments

cv.ob	output of the CVrankSurv_fct
data	same list used as input in CVrankSurv_fct
file	name of the pdf file to be created
...	other arguments, not used know

riskscore_fct	<i>Main function of SurvRank.</i>
---------------	-----------------------------------

Description

Main input function for SurvRank.

Usage

```
riskscore_fct(cv.ob, data, th = 0.5, surv.tab = c(0.5), f = NA,
  fix.var = NA, list.t = "weighted", ncl = 1, plt = F, ...)
```

Arguments

cv.ob	output of the CVrankSurv_fct
data	same list used as input in CVrankSurv_fct
th	Defaults to 0.5. Threshold of used features. th=0.5 majority vote approach
surv.tab	Defaults to c(0.5). Calculates for selected features survival curves. surv.tab determines quantiles of predictions.
f	Defaults to NA. ranking approach function. One of fsSurvRankConc, fsSurvRankGlmnet, fsSurvRankRf, fsSurvRankBoost, fsSurvRankCox, fsSurvRankRandCox, fsSurvRankRpart, fsSurvRankWang or NA, no calculation
fix.var	Defaults to NA. not NA, fixed number of features is calculated
list.t	Defaults to "weighted". Which toplist should be chosen? Possible choices are "weighted", "unweighted", "rank", "top1se", "cluster" or "final"
ncl	Defaults to 1. Number of clusters for parallel execution.
plt	Default=F. Should plot of survival curves be generated?
...	arguments that can be passed to underlying functions, not used now

Details

details to follow

Value

Output of the riskscore_fct, basically a list containing the following elements

selnames	toplist of features that have been chosen
fixR	Matrix of survival AUCs with fixed number of features, but not fixed features!! (could also be calculated before)
model	cox model output for selected features, according to list.t
aic	AIC criterion of cox model
sum.model	summary object of the fitted cox model

concordance	concordance measure of fitted cox model
sfit	survfit object of the cox model)
pfit	predictions of the cox model (fitted values)
sfit.tab	survfit object according to surv.tab separation
sfit.cox	Cox model on the groups generated by surv.tab
sfit.diff	surfdiff: Tests if there is a difference between two or more survival curves using the G-rho family of tests, or for a single curve against a known alternative

Additionally two plots are generated: if f is not NA, a boxplot of the survival AUCs, averaged for cross-validation iterations. The second plot shows the resulting survival curves according to surv.tab.

Examples

```
## Simulating a survival data set
N=100; p=10; n=3
x=data.frame(matrix(rnorm(N*p),nrow=N,p))
beta=rnorm(n)
mx=matrix(rnorm(N*n),N,n)
fx=mx[,seq(n)]*%beta/3
hx=exp(fx)
ty=rexp(N,hx)
tcens=1-rbinom(n=N,prob=.3,size=1)
y=Surv(ty,tcens)
data=list()
data$x<-x; data$y<-y
out<-CVrankSurv_fct(data,2,3,3,fs.method="cox.rank")
## Using the weighted toplist
risk<-riskscore_fct(out,data,list.t="weighted")
## Selected names
risk$selnames
```

risk_newdat

Main function of SurvRank.

Description

Main input function for SurvRank.

Usage

```
risk_newdat(dat_new, sel_names, dat_old, cv.out = 10, c.time = NA,
  detail = NA, plot = F, surv.tab = c(0.5), mcox = T)
```

Arguments

dat_new	a new data set that is not used for the model building but only for prediction
sel_names	the variables that were selected (from riskscore_fct) (see CVrankSurv_fct)
dat_old	the data set used to fit the survival model
cv.out	number of cross-validation folds for the prediction
c.time	as defined in UnoCsurvAUC time; a positive number restricting the upper limit of the time range under consideration
detail	TRUE do the prediction and Uno's C-Statistic computation for the models using 1:sel_names variables FALSE only save the statistics for the different cross validation folds
plot	TRUE do a plot of the survival curves FALSE no plot
surv.tab	Defaults to c(0.5). Calculates for selected features survival curves. surv.tab determines quantiles of predictions.
mcox	TRUE a cox model is fitted FALSE a Cox model with ridge penalty using cv.out cross-validation folds is fitted

Details

details to follow

Value

Output of the risk_newdat, basically a list containing the following elements

unocv	Matrix of censoring-adjusted C-statistic by Uno et al. for the different cross-validation folds and if detail=T as well for different number of variables
unoi	if detail=T Vector of censoring-adjusted C-statistic by Uno et al. for the different number of variables, if detail=FALSE it corresponds to uno_new
rs	model prediction for the new data set
sfit.tab	survfit object according to surv.tab separation
sfit.diff	surfdiff: Tests if there is a difference between two or more survival curves using the G-rho family of tests, or for a single curve against a known alternative
model	model output for dat_old and using the variables given by sel_names
uno_new	the censoring-adjusted C-statistic by Uno et al. using the prediction for dat_new

Additionally if plot is T, the survival curves given by sfit.tab are plotted

Examples

```
## Simulating a survival data set
N=100; p=10; n=3
x=data.frame(matrix(rnorm(N*p),nrow=N,p))
beta=rnorm(n)
mx=matrix(rnorm(N*n),N,n)
fx=mx[,seq(n)]%*%beta/3
hx=exp(fx)
```

```

ty=rexp(N,hx)
tcens=1-rbinom(n=N,prob=.3,size=1)
y=Surv(ty,tcens)
data=list()
data$x<-x; data$y<-y
## CV object
out<-CVrankSurv_fct(data,2,3,3,fs.method="cox.rank")
## The variables selected from the \code{\link{riskscore_fct}}
selected<-riskscore_fct(out,data,list.t="weighted")$selnames
## Applying the risk_newdat function
x=data.frame(matrix(rnorm(N*p),nrow=N,p))
beta=rnorm(n)
mx=matrix(rnorm(N*n),N,n)
fx=mx[,seq(n)]%*%beta/3
hx=exp(fx)
ty=rexp(N,hx)
tcens=1-rbinom(n=N,prob=.3,size=1)
y=Surv(ty,tcens)
data_new=list()
data_new$x<-x; data_new$y<-y
risk<-risk_newdat(data_new,selected,data)

```

weighting_fct

Weighting function for selected features

Description

Weighting function for selected features, according to the performance in the outer loop

Usage

```
weighting_fct(out.mat, t.times, cv.out, out.sc, n1)
```

Arguments

out.mat	matrix of dimension t.times*folds times features according to training set selections. 0/1 matrix: 1 (0) if the feature was (not) selected in the run
t.times	number of times the cross-validation should be repeated
cv.out	number of folds in outer cross validation loop (for estimation of the predictive accuracy)
out.sc	matrix of t.times (of repeated cross validation) times folds (number of outer CV folds) of prediction accuracy evaluations
n1	parameter for functions of two different weighting functions. Deviation from average performance leading to double weight of selected coefficients.

Index

*Topic **SurvRank**

- CVrankSurv_fct, [2](#)
- fin_surv_model_fct, [4](#)
- fsSurvRankBoost, [5](#)
- fsSurvRankConc, [5](#)
- fsSurvRankCox, [6](#)
- fsSurvRankGlmnet, [6](#)
- fsSurvRankRandCox, [7](#)
- fsSurvRankRf, [7](#)
- fsSurvRankRpart, [8](#)
- fsSurvRankWang, [8](#)
- glmnetRank, [9](#)
- plot_CVsurv, [9](#)
- risk_newdat, [11](#)
- riskscore_fct, [10](#)
- weighting_fct, [13](#)

CVrankSurv_fct, [2](#), [4](#), [9](#), [10](#), [12](#)

fin_surv_model_fct, [4](#)
fsSurvRankBoost, [5](#)
fsSurvRankConc, [5](#)
fsSurvRankCox, [6](#)
fsSurvRankGlmnet, [6](#)
fsSurvRankRandCox, [7](#)
fsSurvRankRf, [7](#)
fsSurvRankRpart, [8](#)
fsSurvRankWang, [8](#)

glmnetRank, [9](#)

plot_CVsurv, [9](#)

risk_newdat, [11](#)
riskscore_fct, [10](#)

Surv, [2](#)

weighting_fct, [2](#), [13](#)