Package 'SurvELM'

May 21, 2018

Type Package

Title Survival Anlalysis with Extreme Learning Machines

Version 0.9.0

Author Hong Wang

Maintainer Hong Wang <wh@csu.edu.cn>

Description An R package for Survival Anlalysis with Extreme Learning Machines.

License GPL (>= 2)

Imports Rcpp,survival,RcppNumerical,glmnet,CoxBoost

Depends Matrix,mboost

LinkingTo Rcpp,RcppArmadillo,RcppEigen, RcppNumerical

RoxygenNote 6.0.1

NeedsCompilation yes

R topics documented:

ELMBJ 2
ELMBJEN 3
ELMCox 4
ELMCoxBoost 6
ELMCoxEN 7
ELMmboost 8
predict.ELMBJ 10
predict.ELMBJEN 11
predict.ELMCox 12
predict.ELMCoxBoost 13
predict.ELMCoxEN 14
predict.ELMmboost 15
survFit.ELMmboost 16

Index **18**

ELMBJ *SurvELM ELMBJ*

Description

A Kernel Extreme Learning Machine Using the Buckley-James estimator

Usage

```
ELMBJ(x, y, Regularization_coefficient, kerneltype = 2, Kernel_para = c(2,
  1))
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
Regularization_coefficient	Ridge or Tikhonov regularization parameter. Default value for ELMBJEN is 10000. It need be set by the user here when using a single base ELM survival model. Also known as C in the ELM paper.
kerneltype	Type of kernel matrix. kerneltype=1,a RBF kernel;kerneltype=2 , a linear kernel;kerneltype=3 ,a polynomial kernel;kerneltype=4, a sigmoid kernel.
Kernel_para	Parameters for different types of kernels. A single value for kerneltype=1 or 2. A vector for kerneltype=3 or 4.

Value

List of returned values

trainMSE	Mean Square Error(MSE) on training data.
newy	Esitmated survival times of training data by the Buckley-James estimator.
outputWeight	Weights of the output layer in ELM.

Author(s)

Hong Wang

References

- Hong Wang et al (2017). A Survival Ensemble of Extreme Learning Machine. Applied Intelligence, DOI:10.1007/s10489-017-1063-4.

See Also

ELMBJEN

Examples

```
set.seed(123)
```

```
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
#A kernel ELM base model
kerelmsurv=ELMBJ(trset[,-rii],Surv(trset[,rii[1]],trset[,rii[2]]))
#The traing MSE
tr_mse=kerelmsurv$trainMSE
#New survival times imputed for training data
y_impute=kerelmsurv$newy
```

ELMBJEN *SurvELM ELMBJEN*

Description

A Survival Ensemble of Extreme Learning Machine Using the Buckley-James estimator

Usage

```
ELMBJEN(x, y, mtry = floor(sqrt(ncol(x))), trlength = 100,
  Regularization_coefficient = 10000, Kernel_type = "lin_kernel",
  Kernel_para = c(2, 1))
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
mtry	The number of covariates(predictor variables) used in each base ELM model. Default is the square root of the number of all avaibable covariates.
trlength	The ensemle size (the number of base ELM survival models). Default is 100.
Regularization_coefficient	Ridge or Tikhonov regularization parameter. Default is 10000. Also known as C in the ELM paper.
Kernel_type	Type of kernel matrix. Currently four options avaibable. "RBF_kernel",a RBF kernel;"lin_kernel" , a linear kernel;poly_kernel ,a polynomial kernel;sigmoid_kernel, a sigmoid kernel. Default is "lin_kernel".
Kernel_para	Parameters for different types of kernels. A single value for RBF and linear kernels. A vector for polynomial and sigmoid kernels and progam stops if only a single value is supplied. However, if the vector of values is supplied in the cases of RBF and liner kernels, only the first value will be used. Default is a vector value "c(2,1)"

Value

Object of class ELMSurvEN with elements

elmsurvfit	A list of base models ELMBJ of size trlength. To retrieve a particular base model: use elmsurvfit[[i]], whe
colindexes	Covaraite subspace index.
trlength	Number of bases models trained.

Author(s)

Hong Wang

References

- Hong Wang et al (2017). A Survival Ensemble of Extreme Learning Machine. Applied Intelligence, DOI:10.1007/s10489-017-1063-4.

See Also

ELMBJ

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
## Survival Ensemble of ELM  with default settings
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMBJEN(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
# Get the 1th base model
firstbasemodel=elmsurvmodel$elmsurvfit[[1]]
```

ELMCox *SurvELM ELMCox*

Description

A Regularized Cox Extreme Learning Machine Model

Usage

```
ELMCox(x, y, Kernel_type = "lin_kernel", Kernel_para = c(2, 1), ...)
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
Kernel_type	Type of kernel matrix. Currently four options avaibable. "RBF_kernel",a RBF kernel;"lin_kernel" , a linear kernel;poly_kernel ,a polynomial kernel;sigmoid_kernel, a sigmoid kernel. Default is "lin_kernel".
Kernel_para	Parameters for different types of kernels. A single value for RBF and linear kernels. A vector for polynomial and sigmoid kernels and progam stops if only a single value is supplied. However, if the vector of values is supplied in the cases of RBF and liner kernels, only the first value will be used. Default is a vector value "c(2,1)".
...	Additional arguments for glmnet.

Value

Object of class ELMCox with elements

elmcox	A glmnet type model. See glmnet for details.
trainx	Training data covariates.
kerneltype	Type of kernel matrix used in training. kerneltype=1,a RBF kernel;kerneltype=2 , a linear kernel;kernelty
Kernel_para	Parameters used in training. A single value for kerneltype=1 or 2. A vector for kerneltype=3 or 4.

Author(s)

Hong Wang

See Also

mboost

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
# Default with lasso penalty
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
# with ridge penalty and RBF kernel, alpha has the same meaning as in glmnet
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]],
trset[,rii[2]]),Kernel_type="RBF_kernel",Kernel_para=c(2,1),alpha=0)
# with elastic net penalty
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]),alpha=0.5)
```

```
#The predicted linear predictor
testprelin=predict(elmsurvmodel,teset[,-c(rii)],type="link")
#The predicted  relative-risk
testpreres=predict(elmsurvmodel,teset[,-c(rii)],type="response")
```

ELMCoxBoost *SurvELM ELMCoxBoost*

Description

An Extreme Learning Machine Cox Model with Likelihood Based Boosting

Usage

```
ELMCoxBoost(x, y, Kernel_type = "lin_kernel", Kernel_para = c(2, 1), ...)
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
Kernel_type	Type of kernel matrix. Currently four options avaibable. "RBF_kernel",a RBF kernel;"lin_kernel" , a linear kernel;poly_kernel ,a polynomial kernel;sigmoid_kernel, a sigmoid kernel. Default is "lin_kernel".
Kernel_para	Parameters for different types of kernels. A single value for RBF and linear kernels. A vector for polynomial and sigmoid kernels and progam stops if only a single value is supplied. However, if the vector of values is supplied in the cases of RBF and liner kernels, only the first value will be used. Default is a vector value "c(2,1)".
...	Additional arguments for CoxBoost.

Value

Object of class ELMCoxBoost with elements

elmcoxboost	A CoxBoost model. See CoxBoost for details.
trainx	Training data covariates.
kerneltype	Type of kernel matrix used in training. kerneltype=1,a RBF kernel;kerneltype=2 , a linear kernel;kernelty
Kernel_para	Parameters used in training. A single value for kerneltype=1 or 2. A vector for kerneltype=3 or 4.

Author(s)

Hong Wang

See Also

CoxBoost

Examples

```
set.seed(123)
library(SurvELM)
library(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMCoxBoost(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
#THE predicted linear predictor
testpre=predict(elmsurvmodel,teset[,-c(rii)])
#The predicted cumulative incidence function
testprecif=predict(elmsurvmodel,teset[,-c(rii)],type="CIF")
# The predicted partial log-likelihood
testprellk=predict(elmsurvmodel,teset[,-c(rii)],newtime=teset[,rii[1]],
newstatus=teset[,rii[2]],type="logplik")
uniquetimes=sort(unique(trset$time))
# The predicted probability of not yet having had the event at the time points given in times
testprerisk=predict(elmsurvmodel,teset[,-c(rii)],times=uniquetimes,type="risk")
```

ELMCoxEN *SurvELM ELMCoxEN*

Description

An Ensemble of Regularized Cox Extreme Learning Machine Model

Usage

```
ELMCoxEN(x, y, mtry = floor(sqrt(ncol(x))), enlen = 100,
  Kernel_type = "lin_kernel", Kernel_para = c(2, 1), ...)
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
mtry	Number of covariates within the subset tp build a base ELMCox model.
enlen	Number of base models within the ensemble. Default is 100.
Kernel_type	Type of kernel matrix. Currently four options avaibable. "RBF_kernel",a RBF kernel;"lin_kernel" , a linear kernel;poly_kernel ,a polynomial kernel;sigmoid_kernel, a sigmoid kernel. Default is "lin_kernel".
Kernel_para	Parameters for different types of kernels. A single value for RBF and linear kernels. A vector for polynomial and sigmoid kernels and progam stops if only a single value is supplied. However, if the vector of values is supplied in the cases of RBF and liner kernels, only the first value will be used. Default is a vector value "c(2,1)".
...	Additional arguments for glmnet.

Value

Object of class ELMmboost with elements

elmcoxfit	A list of base ELMCox models of of size enlen.To retrieve a particular base model: use elmcoxfit[[i]], whe
trainx	Training data covariates.
kerneltype	Type of kernel matrix used in training. kerneltype=1,a RBF kernel;kerneltype=2 , a linear kernel;kernelty
Kernel_para	Parameters used in training. A single value for kerneltype=1 or 2. A vector for kerneltype=3 or 4.

Author(s)

Hong Wang

See Also

ELMCox

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
# with ridge penalty and RBF kernel, alpha has the same meaning as in glmnet
elmsurvmodel=ELMCoxEN(x=trset[,-rii],y=Surv(trset[,rii[1]],trset[,rii[2]]),
enlen=10,Kernel_type="RBF_kernel",Kernel_para=c(2,1),alpha=0)
#The second base model
fit2=elmsurvmodel$elmcoxfit[[2]]
#The predicted linear predictor
testprelin=predict(elmsurvmodel,teset[,-c(rii)],type="link")
#The predicted  relative-risk
testpreres=predict(elmsurvmodel,teset[,-c(rii)],type="response")
```

ELMmboost *SurvELM ELMmboost*

Description

An Extreme Learning Machine Cox Model with Gradient Based Boosting

Usage

```
ELMmboost(x, y, Kernel_type = "lin_kernel", Kernel_para = c(2, 1), ...)
```

Arguments

x	The covariates(predictor variables) of training data.
y	Survival time and censored status of training data. Must be a Surv survival object
Kernel_type	Type of kernel matrix. Currently four options avaibable. "RBF_kernel",a RBF kernel;"lin_kernel" , a linear kernel;poly_kernel ,a polynomial kernel;sigmoid_kernel, a sigmoid kernel. Default is "lin_kernel".
Kernel_para	Parameters for different types of kernels. A single value for RBF and linear kernels. A vector for polynomial and sigmoid kernels and progam stops if only a single value is supplied. However, if the vector of values is supplied in the cases of RBF and liner kernels, only the first value will be used. Default is a vector value "c(2,1)".
...	Additional arguments for mboost.

Value

Object of class ELMmboost with elements

elmglmboost	A glmboost model. See mboost for details.
trainx	Training data covariates.
kerneltype	Type of kernel matrix used in training. kerneltype=1,a RBF kernel;kerneltype=2 , a linear kernel;kernelty
Kernel_para	Parameters used in training. A single value for kerneltype=1 or 2. A vector for kerneltype=3 or 4.

Author(s)

Hong Wang

References

- Hong Wang et al (2017). A Survival Ensemble of Extreme Learning Machine. Applied Intelligence, DOI:10.1007/s10489-017-1063-4.

See Also

mboost

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMmboost(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
#THE predicted linear predictor
```

```
testpre=predict(elmsurvmodel,teset[,-c(rii)])
```

`predict.ELMBJ` *SurvELM ELMBJ*

Description

Predicting from A Kernel Extreme Learning Machine Using the Buckley-James estimator

Usage

```
## S3 method for class ELMBJ
predict(object, testx, ...)
```

Arguments

`object`	An object that inherits from class ELMBJEN.
`testx`	A data frame in which to look for variables with which to predict.
`...`	Additional arguments.

Value

List of returned values

`trainMSE`	Mean Square Error(MSE) on training data.
`newy`	Esitmated survival times of training data by the Buckley-James estimator.
`outputWeight`	Weights of the output layer in ELM.
`testpre`	The estimated survival times for `testx` data.

Author(s)

Hong Wang

References

- Hong Wang et al (2017). A Survival Ensemble of Extreme Learning Machine. Applied Intelligence, DOI:10.1007/s10489-017-1063-4.

See Also

ELMBJEN

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
```

```
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
#A kernel ELM base model
kerelmsurv=ELMBJ(trset[,-rii],Surv(trset[,rii[1]],trset[,rii[2]]))
testpre=predict(kerelmsurv,teset[,-c(rii)])
```

`predict.ELMBJEN` *SurvELM predict.ELMBJEN*

Description

Predicting from A Kernel Extreme Learning Machine Ensemble Using the Buckley-James estimator

Usage

```
## S3 method for class ELMBJEN
predict(object, testx, trlength, ...)
```

Arguments

`object`	An object that inherits from class ELMBJEN.
`testx`	A data frame in which to look for variables with which to predict.
`trlength`	Number of based models used for prediction, shouble be less than and equal to the number for training.
`...`	Additional arguments.

Value

produces a vector of predictions or a matrix of predictions

Author(s)

Hong Wang

References

- Hong Wang et al (2017). A Survival Ensemble of Extreme Learning Machine. Applied Intelligence, DOI:10.1007/s10489-017-1063-4.

See Also

`predict.ELMBJ`

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
# with RBF kernel
elmsurvmodel=ELMBJEN(x=trset[,-rii],y=Surv(trset[,rii[1]],trset[,rii[2]]),
trlength=10,Kernel_type="RBF_kernel",Kernel_para=c(2,1))
#The second base model
fit2=elmsurvmodel$elmcoxfit[[2]]
#The predicted survival
testprelin=predict(elmsurvmodel,teset[,-c(rii)])
```

`predict.ELMCox` *SurvELM predict.ELMCox*

Description

Predicting from A Regularized Cox Extreme Learning Machine Model

Usage

```
## S3 method for class ELMCox
predict(object, testx, ...)
```

Arguments

`object`	An object that inherits from class ELMCox.
`testx`	A data frame in which to look for variables with which to predict.
`...`	Additional arguments for `glmnet`.

Value

produces a vector of predictions or a matrix of predictions

Author(s)

Hong Wang

See Also

`predict.glmnet`

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
# Default with lasso penalty
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
# with ridge penalty and RBF kernel, alpha has the same meaning as in glmnet
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]],
trset[,rii[2]]),Kernel_type="RBF_kernel",Kernel_para=c(2,1),alpha=0)
# with elastic net penalty
elmsurvmodel=ELMCox(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]),alpha=0.5)
#The predicted linear predictor
testprelin=predict(elmsurvmodel,teset[,-c(rii)],type="link")
#The predicted  relative-risk
testpreres=predict(elmsurvmodel,teset[,-c(rii)],type="response")
```

`predict.ELMCoxBoost` *SurvELM predict.ELMCoxBoost*

Description

Predicting from An Extreme Learning Machine Cox Model with Likelihood Based Boosting

Usage

```
## S3 method for class ELMCoxBoost
predict(object, testx, ...)
```

Arguments

`object`	An object that inherits from class ELMCoxBoost.
`testx`	A data frame in which to look for variables with which to predict.
`...`	Additional arguments for `CoxBoost`.

Value

produces a vector of predictions or a matrix of predictions

Author(s)

Hong Wang

See Also

predict.CoxBoost

Examples

```
set.seed(123)
library(SurvELM)
library(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMCoxBoost(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
#THE predicted linear predictor
testpre=predict(elmsurvmodel,teset[,-c(rii)])
#The predicted cumulative incidence function
testprecif=predict(elmsurvmodel,teset[,-c(rii)],type="CIF")
# The predicted partial log-likelihood
testprellk=predict(elmsurvmodel,teset[,-c(rii)],newtime=teset[,rii[1]],
newstatus=teset[,rii[2]],type="logplik")
uniquetimes=sort(unique(trset$time))
# The predicted probability of not yet having had the event at the time points given in times
testprerisk=predict(elmsurvmodel,teset[,-c(rii)],times=uniquetimes,type="risk")
```

---

predict.ELMCoxEN *SurvELM predict.ELMCoxEN*

Description

Predicting from an Ensemble of Regularized Cox Extreme Learning Machine Model

Usage

```
## S3 method for class ELMCoxEN
predict(object, testx, enlen, ...)
```

Arguments

object	An object that inherits from class ELMCoxEN.
testx	A data frame in which to look for variables with which to predict.
enlen	Number of based models used for prediction, shouble be less than and equal to the number for training.
...	Additional arguments for glmnet.

Value

produces a vector of predictions or a matrix of predictions

Author(s)

Hong Wang

See Also

predict.glmnet

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
# with ridge penalty and RBF kernel, alpha has the same meaning as in glmnet
elmsurvmodel=ELMCoxEN(x=trset[,-rii],y=Surv(trset[,rii[1]],trset[,rii[2]]),
enlen=10,Kernel_type="RBF_kernel",Kernel_para=c(2,1),alpha=0)
#The second base model
fit2=elmsurvmodel$elmcoxfit[[2]]
#The predicted linear predictor
testprelin=predict(elmsurvmodel,teset[,-c(rii)],type="link")
#The predicted  relative-risk
testpreres=predict(elmsurvmodel,teset[,-c(rii)],type="response")
```

---

predict.ELMmboost *SurvELM predict.ELMmboost*

Description

Predicting from an Extreme Learning Machine Cox Model with Gradient Based Boosting

Usage

```
## S3 method for class ELMmboost
predict(object, testx, ...)
```

Arguments

object	An object that inherits from class ELMmboost.
testx	A data frame in which to look for variables with which to predict.
...	Additional arguments for mboost.

Value

produces a vector of predictions or a matrix of predictions

Author(s)

Hong Wang

See Also

mboost

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMmboost(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
#THE predicted linear predictor
testpre=predict(elmsurvmodel,teset[,-c(rii)])
```

`survFit.ELMmboost` *SurvELM SurvFit.ELMmboost*

Description

An Extreme Learning Machine Cox Model with Gradient Based Boosting

Usage

```
## S3 method for class ELMmboost
survFit(object, newdata = NULL, ...)
```

Arguments

`object`	An ELMmboost object.
`newdata`	An optional data frame in which to look for variables with which to predict the survivor function.
`...`	Additional arguments for mboost.

Value

Object of class `survFit` with elements

`surv`	Estimated survival probabilities at the given time points
`time`	The evaluated given time points.
`n.event`	Number of observed events at given at time points.

See Also

survFit

Examples

```
set.seed(123)
require(SurvELM)
require(survival)
#Lung DATA
data(lung)
lung=na.omit(lung)
lung[,3]=lung[,3]-1
n=dim(lung)[1]
L=sample(1:n,ceiling(n*0.5))
trset<-lung[L,]
teset<-lung[-L,]
rii=c(2,3)
elmsurvmodel=ELMmboost(x=trset[,-rii],y=Surv(trset[,rii[1]], trset[,rii[2]]))
sfit=survFit(elmsurvmodel)
# plot the survival probability
plot(sfit, xlab = "Time", ylab = "Survival Probability")
```

Index

CoxBoost, *6*

ELMBJ, 2, *4*
ELMBJEN, *2*, 3, *10*
ELMCox, 4, *8*
ELMCoxBoost, 6
ELMCoxEN, 7
ELMmboost, 8

mboost, *5*, *9*, *16*

predict.ELMBJ, 10
predict.ELMBJEN, 11
predict.ELMCox, 12
predict.ELMCoxBoost, 13
predict.ELMCoxEN, 14
predict.ELMmboost, 15

survFit.ELMmboost, 16