Package ‘funest’

March 9, 2020

Title Functional Ensemble Survival Tree for Dynamic Prediction

Version 0.0.1.3

Author Yijun Xie [aut, cre],
Shu Jiang [aut],
Graham A. Colditz [aut]

Maintainer Yijun Xie <yijun.xie@uwaterloo.ca>

Description

A fast implementation of functional ensemble survival tree is provided to facilitate dynamic prediction with right-censored data. Multiple time-varying covariates can be accommodated via multivariate principal component analysis. These extracted features along with baseline covariates are nested within the ensemble survival tree where dynamic prediction can be done under user-specified sliding windows. Prediction accuracy measures, Area under the receiver operating characteristic (ROC) curve and Brier score, are provided in this package.

License GPL-3

Encoding UTF-8

LazyData true

KeepSource TRUE

RoxygenNote 7.0.2

Depends R (>= 3.5.0)

Imports MFPCA, funData, ranger, survival, pec, tdROC, prodlim, Rdpack, purrr

RdMacros Rdpack

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-09 10:00:09 UTC

R topics documented:

`funest_fit` *Fitting functional ensemble survival tree model*

Description

The function funest_fit takes a long and a short form of the survival data, among other arguments for a random survival forest, to fit an functional ensemble survival tree model for predicting survival probability.

Usage

```
funest_fit(
  long_train,
  surv_train,
  noftree = 500,
  nofcov = 2,
  split_rule = "maxstat",
  tv_names,
  fv_names,
  nofp = 3,
  t_star,
  t_pred,
  ...
)
```

Arguments

`long_train`	long form of survival data from the training set
`surv_train`	short form of survival data from the training set
`noftree`	number of trees in the random survival forest
`nofcov`	number of covariates selected in each survival tree
`split_rule`	binary splitting rule for random survival forest, default is "maxstat"
`tv_names`	a list of names of time-varying covariates
`fv_names`	a list of names of fixed covariates
`nofp`	number of multivariate principal components
`t_star`	time for the last observed biomarker measurement
`t_pred`	time at prediction
`...`	extra arguments that can be passed to ranger()

Value

A list compose two items. The first item is a list of necessary information for prediction used in funest_pred() function. The second item is the ranger object of the fitted random survival forest.

- misc - a list composed of 1) long_train: long form of survival data from the training set, 2) surv_train: short form of survival data from the training set, 3) fmla: covariates passed into the ensemble survival tree 4) score_names: intermediate names for the covariates 5) nofp: number of multivariate principal components 6) train_data.sub: data frame of all covariates after MFPCA been performed
- rg - functional ensemble survival tree model

References

Jiang S, Xie Y, Colditz GA (2020). “Functional Ensemble Survival Tree: Dynamic Prediction of Alzheimer’s Disease Progression Accommodating Multiple Time-Varying Covariates.” doi: 10.1101/2020.02.17.952994.

Wright MN, Ziegler A (2017). “ranger: A fast implementation of random forests for high dimensional data in C++ and R.” *Journal of Statistical Software*, **77**(i01).

Examples

```
library(funest)
data("long_train")
data("surv_train")
w = funest_fit(long_train, surv_train, tv_names = list("Y1", "Y2", "Y3"), fv_names = list("W"),
 noftree = 10, t_star = 5.5, t_pred = 11)
```

`funest_pred` *Predicting survival probability with time-varing covariates*

Description

The function funest_pred takes the functional ensemble survival tree object from funest_fit() to produce predicted survival probability at user specified t_star and t_pred along with prediction accuracy measures. Must run "predictSurvProb.ranger = predictor_loader()" before calling this function.

Usage

```
funest_pred(
  funest.fit,
  long_test,
  surv_test,
  tv_names,
  fv_names,
  t_star,
  t_pred
)
```

Arguments

`funest.fit`	returned object from funest_fit() function
`long_test`	long form of survival data from the testing set
`surv_test`	short form of survival data from the testing set
`tv_names`	a list of names of time-varying covariates
`fv_names`	a list of names of fixed covariates
`t_star`	time for the last observed biomarker measurement
`t_pred`	time at prediction

Value

A list of three items. The first is a matrix of individual ID and their corresponding predicted survival probability. The second is the estimated Brier score. The third is the estimated area under the ROC curve.

- pred_pb - predicted survival probability at t_pred for each individual conditional on being alive at t_star
- bs - Brier score
- AUC - area under the receiver operating characteristic (ROC) curve

References

Li L, Hu B, Greene T (2015). “A simple method to estimate the time-dependent ROC curve under right censoring.”

Schoop R, Graf E, Schumacher M (2008). “Quantifying the predictive performance of prognostic models for censored survival data with time-dependent covariates.” *Biometrics*, **64**(2), 603–610.

Examples

```
library(funest)
data("long_train")
data("surv_train")
data("long_test")
data("surv_test")
# must run the following line before calling funest_pred()
predictSurvProb.ranger = predictor_loader()
w = funest_fit(long_train, surv_train, tv_names = list("Y1", "Y2", "Y3"),noftree = 10,
 fv_names = list("W"), t_star = 5.5, t_pred = 11)
pred = funest_pred(w, long_test, surv_test, tv_names = list("Y1", "Y2", "Y3"),
 fv_names = list("W"), t_star = 5.5, t_pred = 11)
pred$bs
pred$AUC
```

`long_test` *A sample of long form of testing data*

Description

This contains the long form of the simulated testing data.

Usage

```
data(long_test)
```

Format

data.frame

Examples

```
data(long_test)
```

`long_train` *A sample of long form of training data*

Description

This contains the long form of the simulated training data.

Usage

```
data(long_train)
```

Format

data.frame

Examples

```
data(long_train)
```

predictor_loader *predictor_loader*

Description

An intermediate function for loading the necessary function into .GlobalEnv

Usage

```
predictor_loader()
```

Value

None

Examples

```
# must run the following code before calling funest_pred()
predictSurvProb.ranger = predictor_loader()
```

surv_test *A sample of short form of testing data*

Description

This contains the short form of the simulated testing data.

Usage

```
data(surv_test)
```

Format

data.frame

Examples

```
data(surv_test)
```

`surv_train` *A sample of short form of training data*

Description

This contains the short form of the simulated training data.

Usage

```
data(surv_train)
```

Format

data.frame

Examples

```
data(surv_train)
```

Index