Package 'survAccuracyMeasures'

August 23, 2014

Type Package

Title Estimate accuracy measures for risk prediction markers from survival data

Version 1.2

Date 2014-08-22

Author Yingye Zheng, Tianxi Cai, and Marshall Brown

Maintainer Marshall Brown <mdbrown@fhcrc.org>

Description This package provides a function to estimate the AUC, TPR(c),FPR(c), PPV(c), and NPV(c) for for a specific timepoint and marker cutoff value c using non-parametric and semi-parametric estimators. Standard errors and confidence intervals are also computed. Either analytic or bootstrap standard errors can be computed.

License GPL-2

Depends survival, splines

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp

URL https://github.com/mdbrown/survRpackages

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-08-23 00:46:22

R topics documented:

print.SurvAM *Print an object of class SurvAM–output from survEstMeasures*

Description

S3 method for class "SurvAM"

Usage

```
#S3 method for class ''trtsel''
## S3 method for class 'SurvAM'
print(x, ...)
```

Arguments

x	output from the function survEstMeasures
...	NA

Value

NULL

SimData *Simulated data for package 'SurvAccuracyMeasures'*

Description

Survival data simulated from a Cox proportional hazards model with 20% censoring.

Usage

```
data(SimData)
```

Format

A data frame with 500 observations on the following 3 variables.

`survTime` a numeric vector of survival time

`status` a numeric vector indicating event status, status = 1 for failure, while status = 0 for censored observations

`Y` a numeric vector of values for marker

Examples

```
data(SimData)

head(SimData)
```

`survAM.estimate` *Nonparametric and Semiparametric estimates of accuracy measures for a risk prediction marker from survival data*

Description

This function estimates the AUC, TPR(c), FPR(c), PPV(c), and NPV(c) for for a specific timepoint and marker cutoff value c using semiparametric or nonparametric estimates. Standard errors, and confidence intervals are also computed. Either analytic or bootstrap standard errors can be computed.

Usage

```
survAM.estimate(time, event, marker,
                            data,
                            predict.time,
                            marker.cutpoint = 'median',
                            estimation.method = "IPW",
                            ci.method = "logit.transformed",
                            se.method = "bootstrap",
                            bootstraps = 1000,
                            alpha=0.05)
```

Arguments

`time`	time to event variable
`event`	indicator for the status of event of interest. event = 0 for censored observations, and event = 1 for event of interest.
`marker`	marker variable of interest
`data`	data frame in which to look for input variables.
`predict.time`	numeric value of the timepoint of interest for which to estimate the risk measures
`marker.cutpoint`	numeric value indicating the value of the cutpoint 'c' at which to estimate 'FPR', 'TPR', 'NPV' or 'PPV'. default is `'median'` which takes cutpoint as the marker median.
`estimation.method`	Either "IPW" for non-parametric IPW estimates (default) or "Cox" for semi-parametric estimates that use a Cox proportional hazards model.
`ci.method`	character string of either 'logit.transformed' (default) or 'standard' indicating whether normal approximated confidence intervals should be calculated using logistic transformed values or the standard method.
`se.method`	Method to calculate standard errors for estimates. Options are "bootstrap" (default) or "asymptotic". Asymptotic estimates are based on large sample calculations and will not hold in small samples. Please see referenced papers for more information.

bootstraps	if se.method = 'bootstrap', number of bootstrap replicates to use to estimate the SE.
alpha	alpha value for confidence intervals. (1-alpha)*100 is alpha = 0.05.

Value

a list with components

estimates	point estimates for risk measures
se	standard errors for estimates
CIbounds	bounds for (1-alpha)*100 confidence interval
model.fit	if ESTmethod = "SP", object of type 'coxph' from fitting the model coxph(Surv(time, event)~Y)
cutoff, CImethod, SEmethod, predict.time, alpha	function inputs

References

Liu D, Cai T, Zheng Y. Evaluating the predictive value of biomarkers with stratified case-cohort design. Biometrics 2012, 4: 1219-1227.

Pepe MS, Zheng Y, Jin Y. Evaluating the ROC performance of markers for future events. Lifetime Data Analysis. 2008, 14: 86-113.

Zheng Y, Cai T, Pepe MS, Levy, W. Time-dependent predictive values of prognostic biomarkers with failure time outcome. JASA 2008, 103: 362-368.

Examples

```
data(SimData)

#non-parametric estimates
tmp <- survAM.estimate(time =survTime,
                       event = status,
                       marker = Y,
                       data = SimData,
                       estimation.method = "IPW",
                       predict.time = 2,
                       marker.cutpoint = 0,
                       bootstraps = 50)
tmp
tmp$estimates

#semi-parametric estimates
tmp <- survAM.estimate(time =survTime,
                       event = status,
                       marker = Y,
                       data = SimData,
                       estimation.method = "Cox",
                       predict.time = 2,
                       marker.cutpoint = 0,
                       bootstraps = 50)
```

```
#semi-parametric estimates with asymptotic standard errors
tmp <- survAM.estimate(time =survTime,
                       event = status,
                       marker = Y,
                       data = SimData,
                       estimation.method = "Cox",
                       se.method = "asymptotic",
                       predict.time = 2,
                       marker.cutpoint = 0,
                       bootstraps = 50)
```

Index