Package ‘SurvMetrics’

September 3, 2022

**Type**  Package  
**Title**  Predictive Evaluation Metrics in Survival Analysis  
**Version**  0.5.0  
**Date**  2022-09-03  
**Author**  Hanpu Zhou [aut, cre], Xuewei Cheng [aut], Sizheng Wang [aut], Yi Zou [aut], Hong Wang [aut]  
**Maintainer**  Hanpu Zhou <zhouhanpu@csu.edu.cn>  
**Description**  An implementation of popular evaluation metrics that are commonly used in survival prediction including Concordance Index, Brier Score, Integrated Brier Score, Integrated Square Error, Integrated Absolute Error and Mean Absolute Error. For a detailed information, see (Ishwaran H, Kogalur UB, Blackstone EH and Lauer MS (2008) <doi:10.1214/08-AOAS169>) and (Moradian H, Larocque D and Bellavance F (2017) <doi:10.1007/s10985-016-9372-1>) for different evaluation metrics.  
**License**  GPL (>= 2)  
**Encoding**  UTF-8  
**LazyData**  false  
**Imports**  survminer, stats, survival, MASS, pec, randomForestSRC  
**RoxygenNote**  7.2.1  
**Suggests**  knitr, rmarkdown, caret, ggplot2, ggpubr, testthat (>= 3.0.0)  
**NeedsCompilation**  no  
**URL**  https://github.com/skyee1/SurvMetrics  
**BugReports**  https://github.com/skyee1/SurvMetrics/issues  
**VignetteBuilder**  knitr  
**Config/testthat/edition**  3  
**Repository**  CRAN  
**Date/Publication**  2022-09-03 21:40:23 UTC
R topics documented:

Brier ................................................................. 2
Cindex ............................................................... 4
CindexCR ............................................................ 5
Gt ...................................................................... 6
IAEISE ............................................................... 7
IBS .................................................................. 9
MAE ................................................................. 10
predictSurvProb2survreg ........................................ 11
SDGM1 ............................................................. 12
SDGM2 ............................................................. 13
SDGM3 ............................................................. 14
SDGM4 ............................................................. 15

Index 17

The Brier Score

Description

The Brier Score was proposed by Glenn W. Brier in 1950 which is a proper score function that measures the accuracy of probabilistic predictions, usually used to measure the accuracy of a model fit for survival data. Brier can calculate the value of Brier Score at any timepoint, regardless of whether it is the event time.

Usage

Brier(object, pre_sp, t_star = -1)

Arguments

object object of class Surv created by Surv function or a fitted survival model, including the survival model fitted by coxph, survreg, and rfsrc.

pre_sp If the input of the parameter object is a fitted survival model, this parameter should be a survival dataset on which you want to calculate the Brier Score of the fitted model. Or with an object of class Surv as the parameter object, it should be a vector of predicted values of survival probabilities of each observation in testing set at time t_star.

t_star the timepoint at which the Brier Score you want to calculate. If the input object is a fitted survival model, the timepoint is necessary to be specified at which the survival probability is predicted, and this function will calculate the Brier Score at that moment. If the input object is a survival object, this parameter can be ignored and the value of this parameter will not have any effect on the result of this function.
Details

The Brier Score is the mean square difference between the true classes and the predicted probabilities. So the Brier Score can be thought of as a cost function. Therefore, the lower the Brier Score is for a set of predictions, the better the predictions are calibrated. The Brier Score takes on a value between zero and one, since this is the square of the largest possible difference between a predicted probability and the actual outcome. As we all know, for the cencoring samples, we do not know the real time of death, so the residual cannot be directly calculated when making the prediction. So the Brier Score is widely used in survival analysis.

The Brier Score is a strictly proper score (Gneiting and Raftery, 2007), which means that it takes its minimal value only when the predicted probabilities match the empirical probabilities.

Judging from the sparse empirical evidence, predictions of duration of survival tend to be rather inaccurate. More precision is achieved by using patient-specific survival probabilities and the Brier score as predictions to discriminate future survivors from failures.

Value

the Brier Score at time t_star between the true classes and the predicted probabilities.

Author(s)

Hanpu Zhou <zhouhanpu@csu.edu.cn>

References


Examples

```r
library(survival)
time <- rexp(50)
status <- sample(c(0, 1), 50, replace = TRUE)
pre_sp <- runif(50)
t_star <- runif(1)
Brier(Surv(time, status), pre_sp, t_star)
```
Description

Concordance index is a rank correlation measure between a variable X and a possibly censored variable Y, with event/censoring indicator. In survival analysis, a pair of patients is called concordant if the risk of the event predicted by a model is lower for the patient who experiences the event at a later timepoint. The concordance probability (C-index) is the frequency of concordant pairs among all pairs of subjects. It can be used to measure and compare the discriminative power of a risk prediction models.

Usage

Cindex(object, predicted, t_star = -1)

Arguments

object          object of class Surv created by Surv function or a fitted survival model, including the survival model fitted by coxph, survreg, and rfsrc.
predicted       If the input of the parameter object is a fitted survival model, this parameter should be a survival dataset on which you want to calculate the C-index of the fitted model. Or with an object of class Surv as the parameter object, it should be a vector containing the predicted survival time or probability of each observation.

Details

Pairs with identical observed times, where one is uncensored and one is censored, are always considered usable (independent of the value of tiedOutcomeIn), as it can be assumed that the event occurs at a later timepoint for the censored observation.

For uncensored response the result equals the one obtained with the functions rcorr.cens and rcorrcens from the Hmisc package (see examples).

Value

Estimates of the C-index between the survival time and the predicted result.
**CindexCR**

**Concordance index in the Presence of Competing Risks**

**Description**

The C-index (Concordance index) of the prognostic model in the presence of competing risks according to Marcel, W et al.(2014).

**Usage**

```R
CindexCR(time, status, predicted, Cause_int = 1)
```
Arguments

- **time**: minimum value of deletion time and survival time.
- **status**: the status indicator, for models with competing risks, the status indicator is 0=censored, 1=event at time, 2= competing risks at time.
- **predicted**: a vector of predicted values or the survival time of survival probabilities of each observation.
- **Cause_int**: event type of interest, the default value is 1.

Value

Estimates of the C-index in the presence of competing risks.

Author(s)

HanPu Zhou <zhouhanpu@csu.edu.cn>

References


Examples

```r
time <- c(4, 7, 5, 8)
status <- rep(1, 4)
predicted <- c(3, 5, 7, 10)
Cause_int <- 1
CindexCR(time, status, predicted, Cause_int)
```

The Kaplan-Meier Estimate of the Censoring Distribution

Description

\( G(t) = P(C > t) \) denote the Kaplan-Meier estimate of the censoring distribution which is used to adjust for censoring. \( G(t) \) is used to calculate \( G(t) \) at any timepoint you want.

Usage

\( Gt(object, timepoint) \)
Arguments

object  object of class Surv created by Surv function.
timepoint  any point in time you want to get the Kaplan–Meier estimate of the censoring.

Value

The Kaplan–Meier estimate of the censoring distribution and the value of G(t) is between 0 and 1.

Author(s)

Hanpu Zhou <zhouhanpu@csu.edu.cn>

References


Examples

library(survival)
time <- rexp(50)
status <- sample(c(0, 1), 50, replace = TRUE)
pre_sp <- runif(50)
timepoint <- runif(1)
Gt(Surv(time, status), timepoint)

IAEISE

The Integrate Absolute Error and The Integrate Square Error

Description

Two ways of the continuous-time approach to continuous-time identification based on least-squares and least-absolute errors are proposed. Integrate Absolute Error and Integrate Square Error. To evaluate the performance of survival models methods Lower values of IAE or ISE indicate better performances.

Usage

IAEISE(object, sp_matrix, IRange = c(-2, -1))
Arguments

object

object of class Surv created by Surv function or a fitted survival model, including the survival model fitted by coxph, survreg, and rfsrc.

sp_matrix

a matrix or data.frame of predicted values of survival probabilities for the testing set. rows denote different samples, columns denote different time points, and the values in row i and column j of the matrix denote the predicted survival probability of the ith sample at the time point corresponding to the jth column.

IRange

a vector contains all discrete time points corresponding to the predicted probability in sp_matrix. Or the scale you want to get the IAE and ISE.

Value

Estimates of the Integrate Absolute Error and the Integrate Square Error of the predicted values of survival probabilities.

Author(s)

Hanpu Zhou <zhouhanpu@csu.edu.cn>

References


Examples

library(survival)
library(SurvMetrics)
set.seed(123)
N <- 100
mydata <- SDGM4(N, p = 20, c_step = -0.5)
index.train <- sample(1:N, 2 / 3 * N)
data.train <- mydata[index.train, ]
data.test <- mydata[-index.train, ]
time_interest <- sort(data.train$time[data.train$status == 1])
sp_matrix <- matrix(sort(runif(nrow(data.test) * length(time_interest)),
descending = TRUE ), nrow = nrow(data.test))
object <- Surv(data.test$time, data.test$status)

# a vector for all the distinct time
IAEISE(object, sp_matrix, time_interest)
IBS

The Integration of the Brier Score

Description

IBS is an integrated version of the Brier which is used to calculate the integration of the Brier Score. The Brier Score is the mean square difference between the true classes and the predicted probabilities. Basically, the IBS is an integrated weighted squared distance between the estimated survival function and the empirical survival function. The inverse probability censoring weighting (IPCW) is used to adjust for censoring.

Usage

IBS(object, sp_matrix, IBSrange = c(-2, -1))

Arguments

object: object of class Surv created by Surv function or a fitted survival model, including the survival model fitted by coxph, survreg, and rfsrc.

sp_matrix: a matrix or data.frame of predicted values of survival probabilities for the testing set. Rows denote different samples, columns denote different time points, and the values in entry (i,j) of the matrix denote the predicted survival probability of the ith sample at the time point corresponding to the jth column.

IBSrange: a vector contains all discrete time points corresponding to the predicted probability in sp_matrix. Or the scale you want to get the IBS; and if it is a single point the return value will be the Brier Score at the timepoint.

Details

The percentage of censored observations increases in time, and this will surely affect the dispersion of the empirical Brier Score. The question of how censoring in finite samples acts on the distribution of our measures of inaccuracy is an interesting subject. Our recommendation is to choose t* in a way that censoring is not too heavy (for example, the median follow-up time). We also prefer measures with integrated loss functions since they will reflect inaccuracy over an interval rather than just at one point in time. In addition, the corresponding empirical measures are likely to have lower dispersion, because censored observations contribute their estimated event-free probabilities to the integrand until the censoring occurs.

Value

The integration of the Brier score of the predicted values of survival probabilities on the discrete time points or the time scale of interest to users.
Author(s)
Hanpu Zhou <zhouhanpu@csu.edu.cn>

References

Examples
library(survival)
library(SurvMetrics)
set.seed(123)
N <- 100
mydata <- SDQM4(N, p = 20, c_step = -0.5)
index.train <- sample(1:N, 2/3 * N)
data.train <- mydata[index.train, ]
data.test <- mydata[-index.train, ]

time_interest <- sort(data.train$time[data.train$status == 1])
sp_matrix <- matrix(sort(runif(nrow(data.test) * length(time_interest)), decreasing = TRUE), nrow = nrow(data.test))
object <- Surv(data.test$time, data.test$status)

# the default time points
IBS(object, sp_matrix, time_interest)
# a time range
IBS(object, sp_matrix, c(18:100))

MAE

The Mean Absolute Error

Description
A somewhat naive criterion that is sometimes used consists of simply omitting all censored cases from the data set. For survival analysis problems, the mean absolute error (MAE) can be defined as an average of the differences between the predicted time values and the actual observation time values. Only the samples for which the event occurs are being considered in this metric.

Usage
MAE(object, pre_time)
**Arguments**

- **object**: object of class `Surv` created by `Surv` function.
- **pre_time**: a vector of predicted values of survival time of each observation.

**Details**

Condition: MAE can only be used for the evaluation of survival models which can provide the event time as the predicted target value.

**Value**

the value of the Mean Absolute Error between the survival time and the predicted result.

**Author(s)**

Hanpu Zhou <zhouhanpu@csu.edu.cn>

**References**


**Examples**

```r
library(survival)
time <- rexp(50)
status <- sample(c(0, 1), 50, replace = TRUE)
pre_time <- rexp(50)
MAE(Surv(time, status), pre_time)
```

---

**predictSurvProb2survreg**

*Predicting Survival Probabilities for a `survreg` Object*

**Description**

Function to extract survival probability predictions from `survreg` modeling approach.

**Usage**

```r
predictSurvProb2survreg(object, newdata, time_days)
```
Arguments

object  A model fitted by `survreg` from which to extract predicted survival probabilities.
newdata  A data frame containing predictor variable combinations for which to compute predicted survival probabilities.
time_days  A vector of times in the range of the response variable, e.g. times when the response is a survival object, at which to return the survival probabilities.

Value

A matrix with as many rows as `NROW(newdata)` and as many columns as `length(time_days)`. Each entry should be a probability and in rows the values should be decreasing.

Author(s)

Hanpu Zhou <zhouhanpu@csu.edu.cn>

Examples

```r
library(survival)
set.seed(1234)
mydata <- kidney[, -1]
train_index <- sample(1:nrow(mydata), 0.7 * nrow(mydata))
train_data <- mydata[train_index, ]
test_data <- mydata[-train_index, ]
survregfit <- survreg(Surv(time, status) ~ ., dist = 'weibull', data = train_data)
presb <- predictSurvProb2survreg(survregfit, test_data, c(10, 20))
```

---

**SDGM1**

**Survival Data Generation Method 1**

Description

Survival data generation method. An example of the proportional hazards model where in the Cox model is expected to perform best.

Usage

`SDGM1(N = 200, p = 15, c_mean = 0.4)`

Arguments

- **N**  The sample size of the simulated dataset.
- **p**  The covariate dimension of the simulated dataset.
- **c_mean**  The parameter which is used to control the censoring rate.
**Value**

the simulated dataset

**Author(s)**

Hanpu Zhou <zhouhanpu@csu.edu.cn>

**References**


**Examples**

SDGM1(N = 200, p = 15, c_mean = 0.4)

---

**SDGM2**  
*Survival Data Generation Method 2*

**Description**

Survival data generation method. The dataset represents mild violations of the proportional hazards assumption.

**Usage**

SDGM2(N = 200, p = 15, u_max = 4)

**Arguments**

- **N**  
The sample size of the simulated dataset.

- **p**  
The covariate dimension of the simulated dataset.

- **u_max**  
The parameter which is used to control the censoring rate.

**Value**

the simulated dataset

**Author(s)**

Hanpu Zhou <zhouhanpu@csu.edu.cn>
References


Examples

SDGM2(N = 200, p = 15, u_max = 4)

---

SDGM3

Survival Data Generation Method 3

Description

Survival data generation method. The proportional hazards assumption is strongly violated in this dataset.

Usage

SDGM3(N = 200, p = 15, u_max = 7)

Arguments

- **N**: The sample size of the simulated dataset.
- **p**: The covariate dimension of the simulated dataset.
- **u_max**: The parameter which is used to control the censoring rate.

Value

the simulated dataset

Author(s)

Hanpu Zhou <zhouhanpu@csu.edu.cn>
References

Examples
SDGM3(N = 200, p = 15, u_max = 7)

Description
Survival data generation method. An example of the proportional hazards model where in the Cox model is expected to perform best.

Usage
SDGM4(N = 200, p = 15, c_step = 0.4)

Arguments
N
The sample size of the simulated dataset.
p
The covariate dimension of the simulated dataset.
c_step
The parameter which is used to control the censoring rate.

Value
the simulated dataset

Author(s)
Hanpu Zhou <zhouhanpu@csu.edu.cn>
References


Examples

SDGM4(N = 200, p = 15, c_step = 0.4)
Index

Brier, 2
Cindex, 4
CindexCR, 5
Gt, 6
IAEISE, 7
IBS, 9
MAE, 10
predictSurvProb2survreg, 11
SDGM1, 12
SDGM2, 13
SDGM3, 14
SDGM4, 15