Package 'greyzoneSurv'

May 19, 2015

Type Package

Title Fit a Grey-Zone Model with Survival Data

Version 1.0

Date 2015-05-18

Author Pingping Qu and John Crowley

Maintainer Pingping Qu <pingpingq@crab.org>

Description Allows one to classify patients into low, intermediate, and high risk groups for disease progression based on a continuous marker that is associated with progression-free survival. It uses a latent class model to link the marker and survival outcome and produces two cutoffs for the marker to divide patients into three groups. See the References section for more details.

License GPL-3

Depends stats4, survival, Hmisc, survAUC

NeedsCompilation no

Repository CRAN

Date/Publication 2015-05-19 09:25:40

R topics documented:

greyzoneSurv-package *Fit a Grey-Zone Model with Survival Data*

Description

Allows one to classify patients into low, intermediate, and high risk groups for disease progression based on a continuous marker that is associated with progression-free survival. It uses a latent class model to link the marker and survival outcome and produces two cutoffs for the marker to divide patients into three groups. See the References section for more details.

Details

To fit the grey-zone model, one would need to call the functions in the order of `em.func`, `cov.func`, and `greyzone.func`.

The package also provides a function `bestcut2` to fit a 2-group model, that is, it will find an optimal cutoff of the marker to divide patients into high and low 2 risk groups. Plus there is a function `genSurvData` to generate survival data with a fixed censoring rate.

Author(s)

Pingping Qu and John Crowley

Maintainer: Pingping Qu <pingpingq@crab.org>

References

Pingping Qu, Bart Barlogie and John Crowley (2015) "Using a Latent Class Model to Refine Risk Stratification in Multiple Myeloma" (under review)

bestcut2 *Find an Optimal Cutoff for the 2-group Model*

Description

This function uses a brute force method to search for the best cutoff value for a marker based on the log rank test to divide patients into high and low risk groups given survival data.

Usage

```
bestcut2(data, stime, sind, var, leave = 20)
```

Arguments

data	A data frame or numerical matrix
stime	A character string that tells the column name for survival time in the data
sind	A character string that tells the column name for censoring indicator in the data
var	A character string that tells the column name for marker values in the data
leave	Minimum number of patients in the resulting high and low risk groups

Value

It returns a data frame with the input data as well as the final optimal high and low risk groupings saved in the column bestcut2 (1=high risk and 0=low risk). Additionally, it has columns such as the cutoff value for the marker, the chi-square statistics and the log rank p values for testing equality of survival in the resulting high and low risk groups from using each possible marker value as cutoff.

Examples

```
## Use the package data "mydata" to fit the 2-group model
data(mydata)
res=bestcut2(data=mydata, stime='time', sind='event', var='x')
table(res[,'bestcut2'])

#compare the true groupings and that from the 2-group model
table(res[,c('xhigh', 'bestcut2')])
```

---

cox.summary *Summarize after Fitting Cox regression*

Description

A wrapper function to summarize 2-group and grey-zone (3-group) models with R^2 and c-index after fitting Cox regression.

Usage

```
cox.summary(stime, sind, var)
```

Arguments

stime	A nx1 vector of survival time
sind	A nx1 vector of censoring indicator
var	A nx1 vector of risk groupings to correlate with survival. For the 2-group and 3-group models, it is a categorical vector

Value

It returns a vector with components such as the p value from fitting a Cox regression model, AIC (Akaike information criterion), and the c-index [1, 2], R^2 from the XO method [3], the OXS method [4] and the Nagelkerke method.

References

Harrell F, Califf R, Pryor D, Lee K and Rosati R. Evaluating the yield of medical tests. *JAMA: The Journal of the American Medical Association* 1982; 247(18):2543-2546.

Harrell F, Lee K, Califf R, Pryor D and Rosati R. Regression modeling strategies for improved prognostic prediction. *Statistics in Medicine* 1984; 3(2):143-152.

Xu R and O'Quigley J. A R^2 type measure of dependence for proportional hazards models. *Nonparametric Statistics* 1999; 12:83-107.

O'Quigley J, Xu R and Stare J. Explained randomness in proportional hazards models. *Statistics in Medicine* 2005; 24:479-489.

See Also

greyzone.func

Examples

```
#See Examples for greyzone.func in this package.
```

---

genSurvData *Simulate Survival Outcome Given Marker Values*

Description

This function simulates survival outcome with a fixed censoring rate based on a Weibull distribution given input values such as study recruitment period, patient marker values, their true risk groupings (1=high risk and 0=low risk), and true regression coefficients.

Usage

```
genSurvData(n, recruitment.yrs=2, baseline.hazard=365.25*5, shape=1, censoring.rate=0,
            beta.continuous, beta.binary=0, x, xhigh, ran.seed)
```

Arguments

n	Sample size
recruitment.yrs	Patient recruitment period in years (default=2)
baseline.hazard	Baseline hazard, which is the mean survival time (in days) when covariates=0 (default=365.25*5 days)

shape	The shape parameter for the Weibull distribution (it is exponential when shape=1)
censoring.rate	Censoring rate
beta.continuous	A true regression coefficient that links the continuous marker values to the survival outcome
beta.binary	A true regression coefficient that links the high risk group to the survival outcome
x	A nx1 vector for the marker values
xhigh	A nx1 vector of 1s and 0s indicating patient true risk identities (1=high risk and 0=low risk)
ran.seed	Seed number for random number generation

Details

The function can be used to generate survival data if you do not have any to try the grey-zone model.

Value

It returns a list with two components: the simulated survival data in days and the final censoring rate (which should be the same as the input censoring rate).

See Also

em.func, cov.func, greyzone.func

Examples

```
## Generate package data called "mydata"
## Simulate high/low risk groupings, continuous marker values for each group, and survival data
## so that the higher maker values correspond to shorter survival.
n=300
censoring.rate=0.3
rate.lrisk=0.7   #rate of low risk
n.lrisk=n*rate.lrisk
n.hrisk=n-n.lrisk
mu=3
beta.continuous=0.5
beta.binary=0.5
ran.seed=1000
set.seed(ran.seed)
x0=rnorm(n.lrisk, 0, 1)   #low risk patients have marker values distributed as Normal(0,1)
set.seed(ran.seed)
x1=rnorm(n.hrisk, mu, 1)   #high risk patients have maker values distributed as Normal(mu,1)
score=c(x0, x1)
score.high=c(rep(0, n.lrisk), rep(1, n.hrisk))
mydata=genSurvData(n=n, censoring.rate=censoring.rate,
                beta.continuous=beta.continuous, beta.binary=beta.binary,
                x=score, xhigh=score.high, ran.seed=ran.seed)$data
dim(mydata)
head(mydata)
```

greyzone.funcs *Fit a Grey-Zone Model with Survival Data*

Description

Find two cutoffs for a marker by fitting a grey-zone model that will define high, intermediate, and low risk groups when the outcome is survival.

Usage

```
em.func(error = 0.001, max.iter = 300, initial.values=NULL, y, delta, x)

cov.func(em.results, y, delta, x)

greyzone.func(cov.results, y, delta, x, plot.logistic=T)
```

Arguments

error	Convergence criterion as largest difference in parameter estimates between two consecutive iterations
max.iter	Maximum number of iterations
initial.values	A list of initial parameters with such components as z, lamda, beta, and gamma with the default being NULL. z is an initial guess of dimension nx1 regarding whether each subject or patient is at high risk (=1) or low risk (=0); lamda is an initial guess about the scale parameter for a Weibull distribution (lamda=1 for an exponential distribution); beta is an initial guess on the regression coefficients which include an intercept and a slope linking the latent class variable and the survival outcome, and gamma is an initial guess on the regression coefficients which include an intercept and a slope linking the latent class variable and the marker values.
y	A nx1 vector of survival time in years
delta	A nx1 vector of censoring indicator
x	A nx2 matrix, with the 1st column being all 1s and the 2nd column being the marker values
em.results	Results from calling em.func
cov.results	Results from calling cov.func
plot.logistic	A logical variable. If T, it will plot the fitted logistic function between the marker values and the fitted probability of high risk for each patient

Details

The package assumes higher marker values correspond to shorter survival times, so it is important to make sure this is the case with your data. If initial.values is NULL, the function will generate initial values automatically based on this assumption.

Value

The function em.func returns a list with components such as parameter estimates and number of iterations from the EM algorithm as well as whether the EM algorithm fitting generated an error.

The function cov.func returns a list with components such as the variance-covariance matrix, standard errors and 95% confidence limits for the parameter estimates estimated from calling em.func. An additional component it returns is the estimated probability of being at high risk for each patient given patient survival and marker data.

The function greyzone.func returns a list with components such as the grey zone cutoffs greyzone.ll and greyzone.ul so that patient will be classified as low risk if marker value is < greyzone.ll, high risk if marker value is >= greyzone.ul, and intermediate risk (or in the grey zone) if marker value is >= greyzone.ll and < greyzone.ul.

See Also

genSurvData, cox.summary

Examples

```
#use the package data "mydata" to fit the grey-zone model in 3 steps
data(mydata)
dim(mydata)
head(mydata)

#~step 1: extract information needed for fitting the model and
#make some initial guesses of some parameter values
n=nrow(mydata)
y=mydata$time/365.25
delta=mydata$event
score=mydata$x
x=cbind(rep(1, n), score)

#~step 2: get EM estimates and variance-covariance matrix
results.em=em.func(initial.values=NULL, y=y, delta=delta, x=x)
is.na(results.em$em.error)
#only if above is true, you should proceed; otherwise try a different set
#of intial values and try calling em.func again
names(results.em)
#after you successfully get an EM solution proceed to get the variance-covariance matrix
results.cov=cov.func(results.em, y=y, delta=delta, x=x)
names(results.cov)

#~step 3: when there are no errors above proceed to calculate the grey zone
results=greyzone.func(results.cov, y, delta, x, plot.logistic=FALSE)
names(results)
!is.na(results$greyzone.ll) & !is.na(results$greyzone.ul)
#only when above is true, you have a grey-zone solution and you can proceed
#sometimes there is not a grey-zone solution even if the EM fitting is successful.
#In that case, it means the grey-zone model is not a good fit for the data.
score3=rep(0, n)
score3[score>=results$greyzone.ll & score<results$greyzone.ul]=1
```

```
score3[score>=results$greyzone.ul]=2
#if you get through steps 1-3, you are done fitting the grey-zone model!

#now you may want to compare with a 2-group model, so fit a 2-group model
res=bestcut2(data=mydata, stime='time', sind='event', var='x')
score2=res[,'bestcut2']

#then compare the 2-group and grey-zone models, but note here we compare on the training data
#ideally we want to compare them on a test data set
cox.2group=cox.summary(stime=mydata$time, sind=mydata$event, var=score2)
cox.3group=cox.summary(stime=mydata$time, sind=mydata$event, var=score3)
cox.2group
cox.3group
fit.2group = survfit(Surv(mydata$time, mydata$event)~score2)
fit.3group = survfit(Surv(mydata$time, mydata$event)~score3)
par(mfrow=c(1,2))
plot(fit.2group, lty=1:2, main='2-group model', las=1,
    ylab='Probability of Survival', xlab='Days from Time 0')
plot(fit.3group, lty=1:3, main='grey-zone model', las=1,
    ylab='Probability of Survival', xlab='Days from Time 0')
```

`mydata` *Package Data*

Description

The data set can be used to test the grey-zone and 2-group models.

Usage

```
data(mydata)
```

Format

The data set is a data frame with 300 rows and 4 columns for patient survival time (time), censoring event (event), marker values (x), and true risk groups (xhigh=1 for high risk and 0 for low risk).

Details

The data were generated based on a Weibull distribution for survival times and normal distributions for marker values in both the high and low risk patients.

Source

The data were generated with the code in the Examples of `genSurvData` in this package.

Examples

```
data(mydata)
```

Index