Package 'ROC632'

June 1, 2012

Type Package

Title Estimation of prognostic or diagnostic capacity of microarray data.

Version 0.2

Date 2012-06-01

Author Y. Foucher <Yohann.Foucher@univ-nantes.fr>

Maintainer Y. Foucher <Yohann.Foucher@univ-nantes.fr>

Description This package computes traditional ROC curves and
time-dependent ROC curves using the coss-validation, the 0.632 and the 0.632+ estimators.

License GPL (>=2)

LazyLoad yes

Depends splines, survival, penalized, survivalROC

Imports splines, survival, penalized, survivalROC

URL www.r-project.org, www.divat.fr

R topics documented:

ROC632-package *Estimation of prognostic capacity of microarray data.*

Description

This package can be used for different bootstrap corrections of overfitting in order to estimate the ROC curves and the time-dependent ROC curves.

Details

Package:	ROC632
Type:	Package
Version:	0.2
Date:	2012-06-01
License:	GPL (>=2)
LazyLoad:	yes

Compute different traditional and time-dependent ROC curve using the coss-validation, the 0.632 and the 0.632+ estimators.

ROC	This function performs the estimations of traditional ROC curves (complete data).
AUC	This function computes the area under ROC curve using the trapezoidal rule based on two vectors of sensitivities and specificities.
boot.ROC	This function performs the estimations of ROC curves (complete data) using bootstrap-based algorithmes for correcting the overfitting.
boot.ROCt	This function performs the estimations of time-dependent ROC curves (right censoring) using bootstrap-based algorithmes for correcting the overfitting.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

R. Danger and Y. Foucher. Time dependent ROC curves for the estimation of true prognostic capacity of microarray data. Statistical Applications in Genetics and Molecular Biology. 2012. In press.

See Also

URL: http://www.divat.fr

AUC *Area under ROC curve from sensitivities and specificities*

Description

This function computes the area under ROC curve using the trapezoidal rule.

Usage

```
AUC(sens, spec)
```

Arguments

sens	A numeric vector with the sensitivities
spec	A numeric vector with the sensitivities

Details

This function computes the area under ROC curve using the trapezoidal rule. The value of the area is directly returned.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

Examples

```
se.temp <- c(0, 0.5, 0.5, 1)
sp.temp <- c(1, 0.5, 0.5, 0)
AUC(se.temp, sp.temp)
```

---

boot.ROC — *Estimation of true diagnostic capacity of microarray data (without censoring) using the cross-validation, the 0.632 and the 0.632+ estimators of ROC curves*

Description

This function performs estimations of ROC curves according to different bootstrap-based approaches. The signature is obtained by using a logistic model with lasso penalty.

Usage

```
boot.ROC(status, features, N.boot,
 precision, fold.cv)
```

Arguments

status	A numeric vector with the indicators of the disease (e.g. 0=disease-free, 1=disease).
features	A matrix with the observed features. The number of raw is the number of individuals (length of the argument status).
N.boot	Number of bootstrap iterations.
precision	The quintiles of the predictor used for computing each point of the ROC curve.
fold.cv	The fold for cross-validation (determination of the tuning parameter of the lasso penalty at each bootstrap iteration).

Details

This function computes ROC curve (without censoring data) based on the 0.632+ estimator. At each bootstrap iteration, a logistic model with lasso penalty is estimated. The value of the tuning parameter is also determined by cross-validation at each iteration based on the training sample. The complete methodology is explained by Danger and Foucher (2012) in the context of incomplete data (right censoring). Nevertheless the application of this method is straightforward when the false positive/negative rates are simply obtained by the corresponding observed proportions.

Value

The function returns a list. AUC is a data frame. The raw(s) represent(s) the value(s) of the prognostic time. train is the mean of the areas obtained by using the individuals included in the bootstrap samples (training). valid is the mean of the areas obtained by using the individuals not included in the bootstrap samples (cross-validation). s632 is the mean of the areas obtained by using the simple 0.632 estimator. p632 is the mean of the areas obtained by using the 0.632+ estimator. ROC.Apparent, ROC.CV, ROC.632 and ROC.632p are 4 data frames in which the false negative and positive rates are presented respectively for the 4 estimators: apparent, bootstrap and cross-validation, bootstrap 0.632 and bootstrap 0.632+. These rates correspond to the threashols in cut.values. Coef is a vector of the regression coefficients obtained in the Cox model with lasso penalty obtained by using all subjects. The value of the tuning parameter is equal to Lambda. This model is contained in the object Model. This object is obtained by using the function penalized() in the R package penalized. Please, look at the corresponding help for more details about the object Model. Finally, the signature represents the prognostic score for each subject, i.e. the sum of the regression multiplied by the value of the features.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

R. Danger and Y. Foucher. Time dependent ROC curves for the estimation of true prognostic capacity of microarray data. Statistical Applications in Genetics and Molecular Biology. 2012. In press.

Examples

```
# import and attach the data example

data(DLBCLpatients)
```

```
data(DLBCLgenes)

# WARNING for this example: we only consider the status at the end of
# the follow-up without considering the time-to-event  (this assumption
# can be acceptable if the follow-up time is the same for all subjects,
# if there is no censoring).

res <- boot.ROC(status=DLBCLpatients$f,
 features=DLBCLgenes, N.boot=5,
 precision=seq(0.01, 0.99, by=0.04),
 fold.cv=5)

# The regression coefficients associated
# with the Cox model with lasso penalty
coefficients(res$Model, "all")
res$Coef

# The distribution of the prognostic score
hist(res$Signature, nclass=30, main="",
 xlab="Observed values of the multivariate signature")

# Illustrations of the ROC curve
plot(res$ROC.Apparent$FPR, 1-res$ROC.Apparent$FNR,
 type="b", pch=1, lty=1,
 ylab="True Positive Rates",
 xlab="False Positive Rates")
lines(res$ROC.CV$FPR, 1-res$ROC.CV$FNR,
 type="b", pch=2, lty=2)
lines(res$ROC.632$FPR, 1-res$ROC.632$FNR,
 type="b", pch=3, lty=3)
lines(res$ROC.632p$FPR, 1-res$ROC.632p$FNR,
 type="b", pch=4, lty=4)
legend("bottomright",
 paste(c("Apparent", "CV", "0.632", "0.632+"),
 "curve (AUC=", round(res$AUC,2), ")"), pch=1:4,
 lty=1:4)
```

`boot.ROCt` *Estimation of true prognostic capacity of microarray data using the cross-validation, the 0.632 and the 0.632+ estimators of time-dependent ROC curves*

Description

This function performs estimations of time-dependent ROC curves (with censoring) according to different bootstrap-based approaches. The signature is obtained by using a Cox model with lasso penalty.

Usage

```
boot.ROCt(times, failures, features, N.boot,
 precision, prop, pro.time, fold.cv)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicators (0=right censored, 1=event).
features	A matrix with the observed features. The number of raw is the number of individuals (length of the arguments times and failures).
N.boot	Number of bootstrap iterations.
precision	The quintiles of the predictor used for computing each point of the time dependent ROC curve.
prop	This is the proportion of the nearest neighbors used in the Akritas estimator. The estimation will be based on 2*lambda (1 lambda on the left and 1 lambda on the right) of the total sample size.
pro.time	The prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times.
fold.cv	The fold for cross-validation (determination of the tuning parameter of the lasso penalty at each bootstrap iteration).

Details

This function computes time-dependent ROC curve with right-censoring data based on the 0.632+ estimator. The Akritas approach (nearest neighbor's estimation) is used for ensuring monotone and increasing ROC curve. The theory was defined by Heagerty, Lumley and Pepe (Biometrics, 2000). At each bootstrap iteration, a Cox model with lasso penalty is estimated. The value of the tuning parameter is also determined by cross-validation at each iteration based on the training sample. The complete methodology is explained by Danger and Foucher (2012).

Value

The function returns a list. AUC is a data frame. The raw(s) represent(s) the value(s) of the prognostic time. train is the mean of the areas obtained by using the individuals included in the bootstrap samples (training). valid is the mean of the areas obtained by using the individuals not included in the bootstrap samples (cross-validation). s632 is the mean of the areas obtained by using the simple 0.632 estimator. p632 is the mean of the areas obtained by using the 0.632+ estimator. ROC.Apparent, ROC.CV, ROC.632 and ROC.632p are 4 data frames in which the false negative and positive rates are presented respectively for the 4 estimators: apparent, bootstrap and cross-validation, bootstrap 0.632 and bootstrap 0.632+. These rates correspond to the threshold in cut.values. Coef is a vector of the regression coefficients obtained in the Cox model with lasso penalty obtained by using all the subject. The value of the tuning parameter is equal to Lambda. This model is contained in the object Model. This object is obtained by using the function penalized() in the R package penalized. Please, look at the corresponding help for more details about the object Model. Finally, the signature represents the prognostic score for each subject, i.e. the sum of the regression multiplied by the value of the features.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

R. Danger and Y. Foucher. Time dependent ROC curves for the estimation of true prognostic capacity of microarray data. Statistical Applications in Genetics and Molecular Biology. 2012. In press.

Examples

```
# import and attach the data example

data(DLBCLpatients)
data(DLBCLgenes)

res <- boot.ROCt(times=DLBCLpatients$t, failures=DLBCLpatients$f,
 features=DLBCLgenes, N.boot=5,
 precision=seq(0.01, 0.99, by=0.04),
 prop=0.02, pro.time=5, fold.cv=5)

# The distribution of the prognostic score
hist(res$Signature, nclass=30, main="",
 xlab="Observed values of the multivariate signature")

# Illustrations of the ROC curve
plot(res$ROC.Apparent$FPR, 1-res$ROC.Apparent$FNR,
 type="b", pch=1, lty=1,
 ylab="True Positive Rates",
 xlab="False Positive Rates")
lines(res$ROC.CV$FPR, 1-res$ROC.CV$FNR,
 type="b", pch=2, lty=2)
lines(res$ROC.632$FPR, 1-res$ROC.632$FNR,
 type="b", pch=3, lty=3)
lines(res$ROC.632p$FPR, 1-res$ROC.632p$FNR,
 type="b", pch=4, lty=4)
legend("bottomright",
 paste(c("Apparent", "CV", "0.632", "0.632+"),
 "curve (AUC=", round(res$AUC,2), ")"), pch=1:4,
 lty=1:4)
```

DLBCLgenes *The data concerning the gene expressions of the DLBCL patients*

Description

A matrix with the 7399 gene expressions of the 240 DLBCL patients.

Usage

```
data(DLBCLgenes)
```

Format

A matrix with 240 observations (raws) with the 7399 genes (columns).

Details

Rosenwald et al. (2002) have evaluated tumor samples from 240 DLBCL patients treated with anthracycline based therapy. Te missing data were replaced according to the mean expression level of the nearest 8 genes.

Source

the data is published at http://llmpp.nih.gov/lymphoma/data.shtml.

References

Rosenwald et al. The use of molecular proffiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. New England Journal of Medicine, 346(25):1937-47, 2002.

Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identiffied by gene expression proffiling. Nature, 403(6769):503-11, 2000.

Examples

```
data(DLBCLpatients)
data(DLBCLgenes)

### Patients survival according to the two subgroups defined by using
### the median of the first gene
plot(survfit(Surv(t, f) ~ I(DLBCLgenes[,1] > median(DLBCLgenes[,1])),
 data = DLBCLpatients), xlab="Survival time (in years)",
 ylab="Patient survival", mark.time=FALSE)
```

DLBCLpatients *The data concerning the clinical information of the DLBCL patients*

Description

A data frame with 240 DLBCL patients. The time-to-event is the time to patient death. This time can be right-censored.

Usage

```
data(DLBCLpatients)
```

Format

A data frame with 240 observations (raws) with the 8 following variables (columns).

`ident` This numeric vector represents the key for patient identication

`t` This numeric vector represents the follow up times (until death or censoring)

`f` This numeric vector represents the failure indicators at the follow-up end (1=death, 0=alive)

Details

Rosenwald et al. (2002) evaluated tumor samples from 240 DLBCL patients treated with anthracycline based therapy. They confirmed the existence of the two DLBCL subgroups described previously, GCB-like and ABC-like. The overall survival was significantly different among the subgroups, with 5-year survivals of 60% for the GCB-like and 35% for ABC-like subgroups. An additional third subtype was described with a 5-year survival of 39%.

Source

The data is published at http://llmpp.nih.gov/lymphoma/data.shtml.

References

Rosenwald et al. The use of molecular proffiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. New England Journal of Medicine, 346(25):1937-47, 2002.

Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identiffied by gene expression proffiling. Nature, 403(6769):503-11, 2000.

Examples

```
data(DLBCLpatients)

### Kaplan and Meier estimation of the patients survival
plot(survfit(Surv(t, f) ~ 1, data = DLBCLpatients),
 xlab="Survival time (in years)", ylab="Patient survival",
 mark.time=FALSE)
```

ROC *Estimation of the traditional ROC curves (without censoring)*

Description

This function performs estimations of ROC curves (without censoring) according to quantitative marker and a binary outcome.

Usage

```
ROC(status, marker, cut.values)
```

Arguments

`status`	A numeric vector with the indicators of the disease (e.g. 0=disease-free, 1=disease).
`marker`	A numeric vector with the values of the quantitative marker.
`cut.values`	The threshold values of the marker for which the coordinates of the ROC are computed.

Details

This function computes a traditional ROC curve (without right-censoring). The false positive and negative rates are obtained by estimating the corresponding proportion

Value

The function returns a list. `cut.values` is the vector of the input threshold values. `TP` and `FP` represent the corresponding false and true positive rates. `AUC` is the area under the curve.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

Examples

```
# import and attach the data example

X <- c(1, 2, 3, 4, 5, 6, 7, 8) # The value of the marker
Y <- c(0, 0, 0, 1, 0, 1, 1, 1) # The value of the binary outcome

ROC.obj <- ROC(status=Y, marker=X, cut.values=sort(X))
plot(ROC.obj$FP, ROC.obj$TP, ylab="True Positive Rates",
 xlab="False Positive Rates", type="s", lwd=2)
```

Index