Package 'ROC632'

December 26, 2013

Type Package

Title Estimation of prognostic or diagnostic capacity of microarray data.

Version 0.5

Date 2013-26-12

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 cross-validation, the 0.632 and the 0.632+ estimators. The 0.632+ estimator of time-dependent ROC curve is useful to estimate the predictive accuracy of prognostic signature based on high-dimensional data. For instance, it outperforms the other approaches, especially the cross-validation solution which is often used. The 0.632+ estimators correct the area under the curve in order to adequately estimate the prognostic capacities regardless of the overfitting level. This package also allows for the construction of diagnostic or prognostic scoring systems (penalized regressions). The methodology is adapted to complete data (penalized logistic regression associated with ROC curve) or incomplete time-to-event data (penalized Cox model associated with time-dependent ROC curve).

License GPL (>=2)

LazyLoad yes

Depends splines, survival, penalized, survivalROC

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

R topics documented:

ROC632-package 2
AUC 3
boot.ROC 3
boot.ROCt 5
DLBCLgenes 8
DLBCLpatients 9
ROC 10

Index **11**

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

Description

This package can be used for proposing prognostic scoring systems based on few explanatories variables or based on thousand features from microarray. The methodology is adapted to complete data (penalized logistic regression associated with ROC curve) incomplete time-to-event data (penalized Cox model associated with ROC time-dependent ROC curve).

Details

Package:	ROC632
Type:	Package
Version:	0.5
Date:	2013-26-12
License:	GPL (>=2)
LazyLoad:	yes

ROC	This function allows to compute traditional ROC curves (complete data) for a binary outcome and a continuous marker.
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 allows the construction of a prognostic or a diagnostic signature (complete data) by using bootstrap-based algorithms for correcting the overfitting.
boot.ROCt	This function allows the construction of a prognostic signature (time-to-event data) by using bootstrap-based algorithms 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 Nov 22;11(6):Article 1.

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 by 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 — *Construction of a diagnostic or prognostic scoring system and estimation of its true diagnostic capacities when the data are complete (without censoring)*

Description

This function allows the construction of a diagnostic or prognostic signature by using a logistic regression with lasso penalty. This function also performs estimations of the corresponding ROC curve according to different bootstrap-based approaches. Patients not included in the bootstrap sample are used to correct the overfitting.

Usage

```
boot.ROC(status, features, N.boot,
 precision, fold.cv, lambda1)
```

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 (equals to the length of the vector status).
N.boot	Number of bootstrap iterations.
precision	The quintiles of the predictor used for computing each point of the ROC curve.
lambda1	The fixed values of the tuning parameters for L1 (lasso). If NULL (default value), its value is obtained by cross-validation for the overall sample and at each bootstrap iteration. The reference approach is to re-estimate the tuning parameter and to select the features at each bootstrap iteration. Nevertheless, the 0.632+ estimator appeared to be robust when the tuning was estimated on the full data set and re-used at each step for the features selection. This assumption is associated to an important time-saving. Nevertheless, the complete re-estimation of the model at each iteration remains less open to criticism.
fold.cv	The fold for cross-validation which is only used if lambda1=NULL. By default, A 5-fold cross-validation is implemented.

Details

This function does not deal with censored data. First, this function returns the results of the penalized logistic regression. By default, all the corresponding parameters (including the tuning parameter obtained by cross-validation which defined the number of variables selected in the linear predictor) are obtained from the total sample. The user can also define the value of the tuning parameter. Second, because the resulting scoring system may be associated with overfitting, internal validation is needed. At each iteration and based on each bootstrap sample, a logistic regression with lasso penalty is estimated. By default, the value of the tuning parameter is also determined by cross-validation on each bootstrap sample. Nevertheless, if lambda1 is defined by the user, the same value is used for all the iterations. The complete methodology is explained by Danger and Foucher (2012) in the context of incomplete data (right censoring). The application of this method is straightforward: the false positive/negative rates are simply obtained by the corresponding observed proportions in the function boot.ROC.

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 thresholds defined in cut.values. Coef is a vector of the regression coefficients obtained in the logistic model with lasso penalty obtained by using all subjects. The value of the tuning parameter is equals 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 Nov 22;11(6):Article 1.

Examples

```
# import and attach the data example

data(DLBCLpatients)
data(DLBCLgenes)

# In this exemple, we only reduce the number
# of features, threasholds and iterations for time-saving

DLBCLgenes <- DLBCLgenes[,1:500] # 500 first features
N.iterations <- 2

# If we define a priori the tuning parameter at 15.

res <- boot.ROC(status=DLBCLpatients$f,
 features=DLBCLgenes, N.boot=N.iterations,
 precision=seq(0.05, 0.95, by=0.30), lambda1=15)

# 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, ylim=c(0,1), xlim=c(0,1),
 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` *Construction of a prognostic scoring system and estimation of its true diagnostic capacities when the data are incomplete (right censoring)*

Description

This function allows the construction of a prognostic signature by using a Cox model with lasso penalty. This function also performs estimations of the corresponding time-dependent ROC curve according to different bootstrap-based approaches.

Usage

```
boot.ROCt(times, failures, features, N.boot,
 precision, prop, pro.time, fold.cv, lambda1)
```

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 (equals to the length of the vectors `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 neighbours 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`.
`lambda1`	The fixed values of the tuning parameters for L1 (lasso). If `NULL` (default value), its value is obtained by cross-validation for the overall sample and at each bootstrap iteration. The reference approach is to re-estimate the tuning parameter and to select the features at each bootstrap iteration. Nevertheless, the 0.632+ estimator appeared to be robust when the tuning was estimated on the full data set and re-used at each step for the features selection. This assumption is associated to an important time-saving. Nevertheless, the complete re-estimation of the model at each iteration remains less open to criticism.
`fold.cv`	The fold for cross-validation which is only used if `lambda1=NULL`. By default, A 5-fold cross-validation is implemented.

Details

This function deals with right-censored data. First, this function returns the results of the penalized Cox model. By default, all the corresponding parameters (including the tuning parameter obtained by cross-validation which defined the number of variables selected in the linear predictor) are obtained from the total sample. The user can also define the value of the tuning parameter. Second, because the resulting scoring system may be associated with overfitting, internal validation is needed. At each iteration and based on each bootstrap sample, a Cox model with lasso penalty is estimated. By default, the value of the tuning parameter is also determined by cross-validation on each bootstrap sample. Nevertheless, if `lambda1` is defined by the user, the same value is used for all the iterations. 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 thresholds defined 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 equals 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 Nov 22;11(6):Article 1.

Examples

```
# import and attach the data example

data(DLBCLpatients)
data(DLBCLgenes)

# In this exemple, we only reduce the number
# of features, threasholds and iterations for time-saving

DLBCLgenes <- DLBCLgenes[,1:500] # 500 first features
N.iterations <- 2

# If we define a priori the tuning parameter at 15.

res <- boot.ROCt(times=DLBCLpatients$t, failures=DLBCLpatients$f,
 features=DLBCLgenes, N.boot=N.iterations, precision=seq(0.05, 0.95, by=0.30),
 prop=0.02, pro.time=5, lambda1=15)

# 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. The 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 profiling 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 identified by gene expression profiling. 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 (rows) with the 8 following variables (columns).

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

`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 tumour 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 profiling 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 profiling. Nature, 403(6769):503-11, 2000.

Examples

```
data(DLBCLpatients)

### Kaplan and Meier estimation of the patient 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

∗Topic **0.632+**
ROC632-package, 2
∗Topic **0.632**
boot.ROC, 3
boot.ROCt, 5
ROC632-package, 2
∗Topic **AUC**
AUC, 3
∗Topic **ROC curve**
boot.ROC, 3
ROC, 10
∗Topic **ROC**
ROC632-package, 2
∗Topic **Rosenwald**
DLBCLgenes, 8
DLBCLpatients, 9
∗Topic **Time-dependent ROC curve**
boot.ROCt, 5
∗Topic **bootstrap**
boot.ROC, 3
boot.ROCt, 5
∗Topic **cross-validation**
ROC632-package, 2
∗Topic **datasets**
DLBCLpatients, 9
∗Topic **diffuse large-b-cell lymphoma**
DLBCLgenes, 8
DLBCLpatients, 9
∗Topic **gene expressions**
DLBCLgenes, 8

AUC, 3

boot.ROC, 3
boot.ROCt, 5

DLBCLgenes, 8
DLBCLpatients, 9

ROC, 10
ROC632-package, 2