Package ‘DET’

March 29, 2020

Type Package

Title Representation of DET Curve with Confidence Intervals

Version 2.0.2

Description Builds both ROC (Receiver Operating Characteristic) and DET (Detection Error Tradeoff) curves from a set of predictors, which are the
results of a binary classification system. The curves give a general vision of the performance of the classifier,
and are useful for comparing performance of different systems.

License GPL-2

Encoding UTF-8

Depends R (>= 3.5.0)

Imports pROC, doParallel, parallel

RoxygenNote 6.1.1

NeedsCompilation no

Author García-Ródenas, Álvaro [aut, cre],
Franco, Manuel [aut],
Vivo, Juana-María [aut],
Fernández-Breis, Jesualdo T. [aut],
Font, Roberto [aut]

Maintainer ``García-Ródenas, Álvaro'' `<alvaro.garcia9@um.es>`

Repository CRAN

Date/Publication 2020-03-29 11:30:02 UTC

R topics documented:

`detc` *DET Curve calculation*

Description

From a dataframe of responses and predictors, the function calculates and plots the DET curves for each pair (response,predictor). In addition, ROC curves can be displayed by activating a boolean argument.

Usage

```
detc(responses, predictors, names = c(""), positive = "", title = "",
  legend = "topright", file, plotROC = FALSE, xlim = c(0.05, 50),
  ylim = c(0.05, 50), col = c("black", "blue", "red", "green",
  "yellow"))
```

Arguments

`responses`	A dataframe of factor, numeric or character vector of responses, typically encoded with 0 (non-target) and 1 (target). By default, the first two values of levels(as.factor(response)) are taken. If only one response is passed, it will be used for calculating all the curves.
`predictors`	A dataframe of numeric vector of the same length than response, containing the predicted value of each observation. An ordered factor is coerced to a numeric.
`names`	Array of strings, containing the name of each pair (response,predictor) that will appear in the legend of the graph.
`positive`	string with the name of the 'positive' class. Default: negative class will be the first string of response levels and positive class the second string.
`title`	Main tile for the graph
`legend`	the location of the leyend in the graph. Could be a single keyword from the list "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center" and NULL. Default: "topright".
`file`	The name of the file where the plot will be saved. If Empty, the DET and ROC curves are plotted by the graphic output.
`plotROC`	Boolean specifying ploting or not the ROC Curve
`xlim`	numeric vector of length 2, giving the x coordinates range.
`ylim`	numeric vector of length 2, giving the y coordinates range.
`col`	Array indicating a specification for the plotting color for DET curve of each classifier.

Value

A list of dataframe, one per classifier. Each dataframe contains the name of the classifier, and the parameters of the DET curve (FPR, FNR and the thresholds used), along with the Equal Error Rate.

Examples

```
n <- 5000
set.seed(12345)
#Predictors with normal distribution
set.seed(1235)
scoreNegative <- rnorm(n, mean = 0.25,sd = 0.125)
set.seed(11452)
scorePositive1 <- rnorm(n, mean = 0.55,sd = 0.125)
set.seed(54321)
scorePositive2 <- rnorm(n, mean = 0.65,sd = 0.125)
set.seed(65987)
scorePositive3 <- rnorm(n, mean = 0.75,sd = 0.125)
response = c(rep(c("target"), times = n),rep(c("nontarget"), times = n))
predictor1 = c(scoreNegative,scorePositive1)
predictor2 = c(scoreNegative,scorePositive2)
predictor3 = c(scoreNegative,scorePositive3)
responses <- data.frame(
  response = response
)
predictors <- data.frame(
  DET1 = predictor1,
  DET2 = predictor2,
  DET3 = predictor3
)
#We can also plot the ROC curves activating logical attribute 'plotROC'
detcurve <- detc(responses,predictors,
                names = names(predictors),
                positive="target",
                title="Example",
                plotROC = TRUE)

#If you want to plot the EER and its CI on the curves:
for (name in names(detcurve)){
    points(qnorm(detcurve[[name]]$EER),qnorm(detcurve[[name]]$EER),
           pch=19,col = detcurve[[name]]$color,lwd=3)
}
```

`detc.CI` *DET Curve calculation with Confidence Interval*

Description

From a dataframe of responses and predictors, the function calculates and plots DET curves for each pair (response,predictor), with a Confidence Interval. In addition, ROC curves can be displayed by activating a boolean argument.

Usage

```
detc.CI(responses, predictors, conf = 0.95, names = c(""),
  positive = "", title = "", legend = "topright", parallel = FALSE,
  ncores = 2, file, plotROC = FALSE, xlim = c(0.05, 50),
  ylim = c(0.05, 50), col = c("black", "blue", "red", "green",
  "yellow"), colbands = c("lightgray", "lightblue", "lightpink",
  "lightgreen", "lightyellow"))
```

Arguments

`responses`	A dataframe of factor, numeric or character vector of responses, typically encoded with 0 (non-target) and 1 (target). By default, the first two values of levels(as.factor(response)) are taken. If only one response is passed, it will be used for calculating all the curves.
`predictors`	A dataframe of numeric vector of the same length than response, containing the predicted value of each observation. An ordered factor is coerced to a numeric.
`conf`	The width of the confidence interval as [0,1]. Default: 0.95 (95% CI).
`names`	Array of strings, containing the name of each pair (response,predictor) that will appear in the legend of the graph.
`positive`	string with the name of the 'positive' class. Default: negative class will be the first string of response levels and positive class the second string.
`title`	Main tile for the graph
`legend`	the location of the leyend in the graph. Could be a single keyword from the list "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center" and NULL. Default: "topright".
`parallel`	if TRUE, the bootstrap is processed in parallel, using parallel backend provided by plyr (foreach).
`ncores`	The number of nodes to be forked for the parallel computation. Default: 2.
`file`	The name of the file where the plot will be saved . If Empty, the DET and ROC curves are plotted by the graphic output.
`plotROC`	Boolean specifying ploting or not the ROC Curve
`xlim`	numeric vector of length 2, giving the x coordinates range.
`ylim`	numeric vector of length 2, giving the y coordinates range.
`col`	Array indicating a specification for the plotting color for DET curve of each classifier.
`colbands`	Array indicating a specification for the plotting color for DET CI bands of each classifier.

Value

A list of dataframe, one per classifier. Each dataframe contains the name of the pair, and the parameters of the DET curve (FPR, the median of FNR and the upper and lower extremes for the CI, and the thresholds used), along with the confidence interval for the Equal Error Rate .

Examples

```
n <- 500
#Predictors with normal distribution
set.seed(1235)
scoreNegative <- rnorm(n, mean = 0.25,sd = 0.125)
set.seed(11452)
scorePositive1 <- rnorm(n, mean = 0.55,sd = 0.125)
set.seed(54321)
scorePositive2 <- rnorm(n, mean = 0.65,sd = 0.125)
response = c(rep(c("target"), times = n),rep(c("nontarget"), times = n))
predictor1 = c(scoreNegative,scorePositive1)
predictor2 = c(scoreNegative,scorePositive2)
responses <- data.frame(
  response = response
)
predictors <- data.frame(
  DET1 = predictor1,
  DET2 = predictor2
)
#Run in parallel for a faster execution (takes about 3-5 min in a 2018 laptop) activating
#logical argument 'parallel'
detcurve <- detc.CI(responses,predictors,
                   names = names(predictors),
                   title = "Example with CI",
                   positive="target",
                   parallel = TRUE)
#If you want to plot the EER and its CI on the curves:
for (name in names(detcurve)){
    points(qnorm(detcurve[[name]]$EER_median),qnorm(detcurve[[name]]$EER_median),
           pch=19,col = detcurve[[name]]$color,lwd=2.5)
    points(qnorm(detcurve[[name]]$EER_lower),qnorm(detcurve[[name]]$EER_lower),
           pch=20,col = detcurve[[name]]$color,lwd=2.5)
    points(qnorm(detcurve[[name]]$EER_upper),qnorm(detcurve[[name]]$EER_upper),
           pch=20,col = detcurve[[name]]$color,lwd=2.5)
}
```

`ovarianCancer` *Ovarian Cancer (NCI PBSII Data)*

Description

The database used correspond to proteomic spectra, generated by mass spectroscopy. This data dates from 6-19-02, and includes 91 controls (Normal) and 162 ovarian cancers. The raw spectral data of each sample contains the relative amplitude of the intensity at each molecular mass/charge (M/Z) identity. There are total 15154 M/Z identities. The intensity values were normalized according to the formula: NV = (V-Min)/(Max-Min) where NV is the normalized value, V$ the raw value, Min the minimum intensity and Max the maximum intensity. The normalization is done over all the 253 samples for all 15154 M/Z identities. After the normalization, each intensity value is to fall within the range of 0 to 1.

Usage

```
data(ovarianCancer)
```

Format

An object of class `"data.frame"`.

Source

ZIP Archive

References

Emanuel F Petricoin et al. (2002) The Lancet 359:572-577 (PubMed)

Examples

```
data(ovarianCancer)
responses <- data.frame(ovarianCancer$response)
predictors <- data.frame(
  n1 = as.numeric(as.numeric(ovarianCancer[[2]])),
  n2 = as.numeric(as.numeric(ovarianCancer[[3]]))
)
names(predictors) = c("Protein 1689","Protein 1737")
detcurve <- detc(responses,predictors,
                 names = names(predictors),
                 positive = "Cancer",
                 title = "Proteomic patterns",
                 plotROC = TRUE)
```

`printDET` *DET Curve information*

Description

From a DET curve list (generated with detc() or detc.CI()), the function shows the different atributtes of each curve with a little description.

Usage

```
printDET(dets)
```

Arguments

`dets`	list of DET curves originated by detc() or detc.CI() functions

`speaker` *Speaker Recognition System (Voxceleb verification test)*

Description

For our experiments, we have used the Voxceleb database, which contains more than one hundred thousand utterances extracted from Youtube interview videos. The database includes training and test sets that can be used for speaker recognition system development and performance evaluation respectively. The testing protocol consists of a list of utterance pairs, with the corresponding target or nontarget, and the task is to detect whether the two utterances belong to the same speaker or to different ones.

Usage

```
data(speaker)
```

Format

An object of class `"data.frame"`.

Source

Web Archive

References

Nagraniy A, Chungy JS, Zisserman A (2017). Proceedings of the Annual Conference of the International Speech Communication Association, 950:2616–2620 (Publication)

Examples

```
data(speaker)
scoresLDA <-speaker$scoresLDA
scoresPLDA <-speaker$scoresPLDA
scoresLDAPLDA <- speaker$scoresLDAPLDA
responses <- data.frame(speaker$keys$V3)
predictors <- data.frame(
  n1 = as.numeric(scoresLDA),
  n2 = as.numeric(scoresPLDA),
  n3 = as.numeric(scoresLDAPLDA)
)
detcurve <- detc(responses,predictors,
                 names = c("LDA+DC","PLDA","LDA+PLDA"),
                 positive = "target",
                 title = "Voxceleb verification test",
                 plotROC = TRUE)
```

Index