Logistic regression and ROC curves

Hugo Varet
2018/10/11

1 Import the dataset:

```r
                   header=TRUE, sep=",")
str(data)
```

```r
## 'data.frame': 714 obs. of 3 variables:
## $ cancer : logi FALSE TRUE FALSE FALSE FALSE FALSE ...
## $ smoker : logi TRUE TRUE FALSE TRUE FALSE TRUE ...
## $ pack_year: int 15 28 0 22 0 6 0 35 0 0 ...
```

2 Plot cancer in function of pack_year:

Basic plot:

```r
plot(cancer ~ pack_year, data=data)
```

But boxplots are more adapted to visualize these data:

```r
boxplot(pack_year ~ cancer, data=data, xlab="Cancer", ylab="# of packs per year")
```
3 Use the `glm()` function to explain the cancer variable with all the other variables:

```r
fit <- glm(cancer ~ pack_year + smoker, data=data, family="binomial")
summary(fit)
```

```
## Call:
## glm(formula = cancer ~ pack_year + smoker, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.6512 -1.0090 -0.4408  1.1052  2.1815
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.282382   0.234697  -9.725  < 2e-16 ***
## pack_year   0.044780   0.009382   4.773  1.82e-06 ***
## smokerTRUE  1.469415   0.291424   5.042  4.60e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 931.88  on 713  degrees of freedom
## Residual deviance: 797.18 on 711  degrees of freedom
## AIC: 803.18
##```
4 Evaluation of the model using the ROC curve and the AUC criteria

Distribution of the values (probabilities) predicted by the logistic model:

```r
summary(fit$fitted.values)
```

```
## Min. 1st Qu.  Median   Mean 3rd Qu.   Max. 
## 0.09259 0.09259 0.39892 0.35854 0.48708 0.88165
```

```r
hist(fit$fitted.values)
```

Histogram of fit$fitted.values

Number of patients having cancer in the data set:

```r
sum(data$cancer)
```

```
## [1] 256
```

We search an optimal threshold such as if the value predicted is superior to it, the model predicts the patient has a cancer.

Number of predicted values superior to 0.4 / 0.45 / 0.5:

```r
sum(fit$fitted.values > 0.4)
```

```
## [1] 345
```

```r
sum(fit$fitted.values > 0.45)
```

```
## [1] 241
```
```r
sum(fit$fitted.values > 0.5)

## [1] 160
```

It seems that a threshold around 0.45 allows having a number of predicted patients with a cancer similar to the number of patients having really a cancer.

```r
threshold <- 0.45
# Number of true positives
sum((fit$fitted.values > threshold) & data$cancer)

## [1] 136
```

# sensitivity = Number of true positives / Number of total (or real) positives
sensitivity <- sum((fit$fitted.values > threshold) & data$cancer) / sum(data$cancer)

# FPR = Number of false positives / Number of total (or real) negatives
# specificity = 1 - FPR
# specificity = Number of true negatives / Number of total (or real) negatives
specificity <- sum((fit$fitted.values <= threshold) & (!data$cancer)) / sum(!data$cancer)

```r
print(sensitivity)

## [1] 0.53125
```

```r
print(specificity)

## [1] 0.7707424
```

With a threshold equal to 0.45 on the fitted probabilities:

- 53% of patients with cancer are correctly classified,
- 77% of patients without cancer are correctly classified.

Note that the goal is to avoid:

- telling a patient without cancer that he has one,
- telling a patient with cancer that he is healthy.

For each probability threshold between 0 an 1, the sensitivity and specificity of the decision rule based on the logistic model can be computed:

```r
sensitivity <- NULL
specificity <- NULL
for (threshold in seq(0, 1, by=0.01)){
  # sensitivity = #True positives / #Total positives
  sens.tmp <- sum((fit$fitted.values > threshold) & data$cancer) / sum(data$cancer)
  sensitivity <- c(sensitivity, sens.tmp)
  # specificity = #True negatives / #Total negatives
  spe.tmp <- sum((fit$fitted.values <= threshold) & (!data$cancer)) / sum(!data$cancer)
  specificity <- c(specificity, spe.tmp)
}
print(sensitivity)
```

```r
## [1] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
## [7] 1.00000000 1.00000000 1.00000000 1.00000000 0.92187500 0.92187500
## [13] 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500
## [19] 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500
## [25] 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500 0.92187500
## [31] 0.92187500 0.92187500 0.91796875 0.90625000 0.89062500 0.85156250
## [37] 0.82031250 0.78515625 0.75781250 0.73828125 0.70312500 0.66796875
## [43] 0.66796875 0.61718750 0.57031250 0.53125000 0.48828125 0.46484375
```
Plotting 1-specificity versus sensitivity for each threshold allows obtaining a ROC (Receiver operating characteristic) curve:

```r
plot(x=1-specificity, y=sensitivity, type="l",main="ROC curve",
     xlab="False Positive Rate (or 1-specificity)",
     ylab="True Positive Rate (or sensitivity)"
)
#Writing each threshold on the ROC curve
eps <- 0.02
text(x=1-specificity-eps, y=sensitivity, labels = as.character(seq(0,1,by=0.01)), cex=0.5)
```
Computing the AUC criteria to evaluate the discriminating power of the logistic model:

```r
library(MESS)

## Loading required package: geepack
## Loading required package: geeM
## Loading required package: Matrix
MESS::auc(x=1-specificity,y=sensitivity)

## [1] 0.7352228
```

If the AUC is not superior to 0.5, it means that the model is not better than a “random” model (i.e. randomly attribute a cancer to the patients).

The closer to 1 the AUC criteria, the more adequate the model is to predict.

You can also use the ROC function of the Epi library to compute the ROC curve:

```r
library(Epi)
ROC(test=fit$fitted.values, stat=data$cancer, plot="ROC")
```
5 Use the predict() function to fit the probability of having cancer for a smoker with 10 pack_years

```r
# create a data frame containing the values of the new patient
ew.patient <- data.frame(smoker=TRUE, pack_year=10)
predict(fit, new.patient, type="response")
```

```r
## 1
## 0.4097089
```

Which is equivalent to computing:

```r
exp((coef(fit)[1] + 10*coef(fit)[2] + coef(fit)[3])) / (1+exp((coef(fit)[1] + 10*coef(fit)[2] + coef(fit)[3])))
```

```r
## (Intercept)
## 0.4097089
```

6 Estimate the odds-ratio of developing cancer when smoking

```r
exp(coef(fit)[3])
```

```r
## smokerTRUE
## 4.34669
```

```r
exp(confint(fit, 3))
```

```r
## Waiting for profiling to be done...
## 2.5 % 97.5 %
```
The odds-ratio between a smoker and a non-smoker is equal to 4.34.

7 Estimate the odd ratio of developing cancer when having smoked 20 pack_years compared to 10

```r
exp((20 - 10) * coef(fit)[2])
```

## pack_year
## 1.564864

```r
exp((20 - 10) * confint(fit, 2))
```

## Waiting for profiling to be done...
## 2.5 %  97.5 %
## 1.307327 1.889704

8 Assumption not satisfied

The main problem is that the two explanatory covariates are not independent: people who do not smoke have a number of packs per year at 0 while people who smoke have a positive value for this variable. The estimations of the coefficients of the model can thus be biased.

```r
boxplot(pack_year ~ smoker, data=data)
```