ICT for Health
Laboratory # 4
ROC - Covid-19 serological tests

Monica Visintin

Politecnico di Torino

2020/21

Table of Contents

Description [1]

Covid-19 can be diagnosed using nasopharyngeal swabs or using serological tests (or using lung CT-scans, see lab #3).
Serological tests are faster, but nasopharyngeal swabs are more reliable (higher sensitivity and specificity).
For some patients nasopharyngeal and two serological tests were used; for the serological tests the marker levels were recorded, for the swab test only the result (positive/negative) was recorded.
Goal of the lab is to set the threshold of the marker above which each serological test is considered positive, using ROC curves, and compare the two serological tests, deciding which should be used.

Table of Contents

Data analysis

The input file is `covid_serological_results.csv` and contains 3 columns:

- `COVID_swab_res` which is the swab result: 0 means negative, 2 means positive, 1 means uncertain result
- `IgG_test1_titre` which is the level of Immunoglobulin G (IgG) in test #1; values range from 2.5 to 314
- `IgG_test2_titre` which is the level of Immunoglobulin G (IgG) in test #2; values range from 0 to 9.71

The dataset includes 879 rows, without missing values. Note that the scale with which IgG is measured is somehow arbitrary and therefore each test will have its own threshold.
We will consider swab test result as ground truth (which might not be correct, since also the swab test has its own sensitivity and specificity), and we will remove from the dataset patients with uncertain (i.e. equal to 1) swab tests.

Read the data

The first lines in the script are the following:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
xx=pd.read_csv("covid_serological_results.csv")
swab=xx.COVID_swab_res.values# results from swab: 0= neg., 1 = unclear, 2=pos.
Test1=xx.IgG_Test1_titre.values
Test2=xx.IgG_Test2_titre.values
# remove uncertain swab tests
ii=np.argwhere(swab==1).flatten()
swab=np.delete(swab,ii)
swab=swab//2
Test1=np.delete(Test1,ii)
Test2=np.delete(Test2,ii)
```

Table of Contents

Find sensitivity

Let us focus on `Test1`. We arbitrarily set the threshold to 20, and we find sensitivity and specificity for this threshold.

```
x=Test1
y=swab
x0=x[swab==0] # test results for healthy patients
x1=x[swab==1] # test results for ill patients
Np=np.sum(swab==1) # number of ill patients
Nn=np.sum(swab==0) # number of healthy patients
thresh = 20 # example of threshold
n1=np.sum(x1>thresh) # number of true positives for the given thresh
sens=n1/Np # sensitivity
n0=np.sum(x0<thresh) # number of true negatives
spec=n0/Nn # specificity
```

Sensitivity is 0.789 and specificity is 0.949.

Find false alarm

ROC (Receiver Operating Curve) is sensitivity versus false alarm:

$$P(T_p|D) \text{ versus } P(T_p|H)$$

Therefore we are not interested in specificity, actually, but false alarm is 1 minus specificity:

$$P(T_p|H) = 1 - P(T_n|H)$$

```
21 FA=1-spec
```

False alarm is 0.05. Therefore, when the threshold is set equal to 20, the point in the ROC curve has abscissa 0.05 and ordinate 0.789. We must find sensitivity and false alarm for all the possible thresholds (of course between the minimum (2.5) and the maximum (314) values of `Test1`.

ROC curve [1]

Instead of selecting uniformly spaced thresholds between 2.5 and 314, it is better to consider the thresholds equal to the possible values taken by `Test1`, from the smallest to the highest, and it is also convenient to write a method/function:

```
1  def findROC(x,y):#
2      """ findROC(x,y) generates data to plot the ROC curve.
3      x and y are two 1D vectors each with length N
4      x[k] is the scalar value measured in the test
5      y[k] is either 0 (healthy person) or 1 (ill person)
6      The output data is a 2D array N rows and three columns
7      data[:,0] is the set of thresholds
8      data[:,1] is the corresponding false alarm
9      data[:,2] is the corresponding sensitivity"""
10     x0=x[y==0] #  test values for healthy patients
11     x1=x[y==1] #  test values for ill patients
12     xss=np.sort(x) #  sort test values to get all the possible  thresholds
13     xs=np.unique(xss) #  remove repetitions
14     if xs[0]>0:
15         xs=np.insert(xs,0,0) # add 0 as first element
16     Np=np.sum(y==1) #  number of ill patients
17     Nn=np.sum(y==0) #  number of healthy patients
18     data=np.zeros((len(xs),3),dtype=float)
```

ROC curve [2]

```
19
20      for i,thresh in enumerate(xs):
21          n1=np.sum(x1>thresh)
22          sens=n1/Np
23          n2=np.sum(x0<thresh)
24          spec=n2/Nn
25          FA=1-spec
26          data[i,0]=thresh
27          data[i,1]=FA
28          data[i,2]=sens
29      return data
```

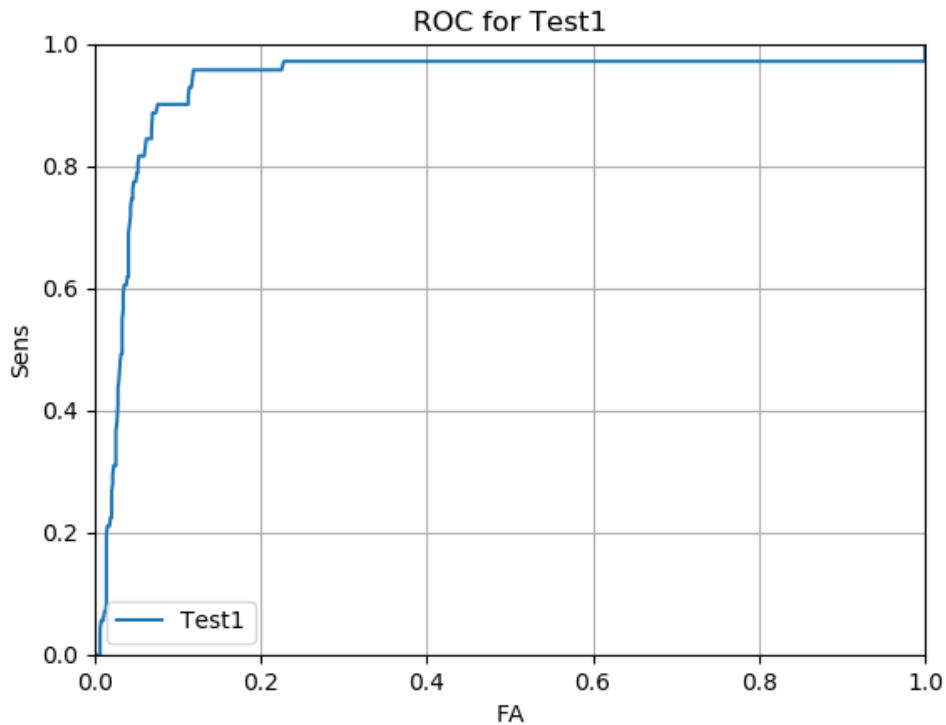
ROC curve [3]

Then the main part of the script becomes

```
plt.close('all')
xx=pd.read_csv("covid_serological_results.csv")
swab=xx.COVID_swab_res.values# results from swab: 0= neg., 1 = unclear, 2=pos.
Test1=xx.IgG_Test1_titre.values
Test2=xx.IgG_Test2_titre.values
swab[swab>=1]=1#
data_Test1=findROC(Test1,swab)
plt.figure()
plt.plot(data_Test1[:,1],data_Test1[:,2],label='Test1')
plt.axis([0,1,0,1])
plt.xlabel('FA')
plt.ylabel('Sens')
plt.grid()
plt.legend()
plt.title('ROC for Test1')
```

ROC curve [4]

The shown ROC curve is the following:



The ideal working point is for FA=0 and sensitivity equal to 1, which is not reached by this test; however the results are not so bad, after all.

Table of Contents

How to set the threshold [1]

There are several ways to set the threshold above which the test is considered positive:

1. The required sensitivity is set, and the associated false alarm (and therefore specificity) is obtained as a consequence.
2. The required false alarm is set, and the associated sensitivity is obtained as a consequence.
3. Equal values of sensitivity and specificity are set (when sensitivity and specificity are both important at the same level).

How to set the threshold [2]

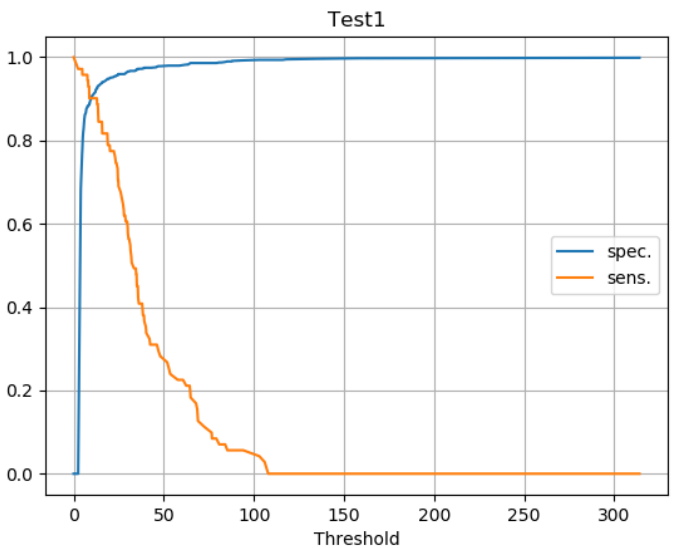
In the COVID-19 case,

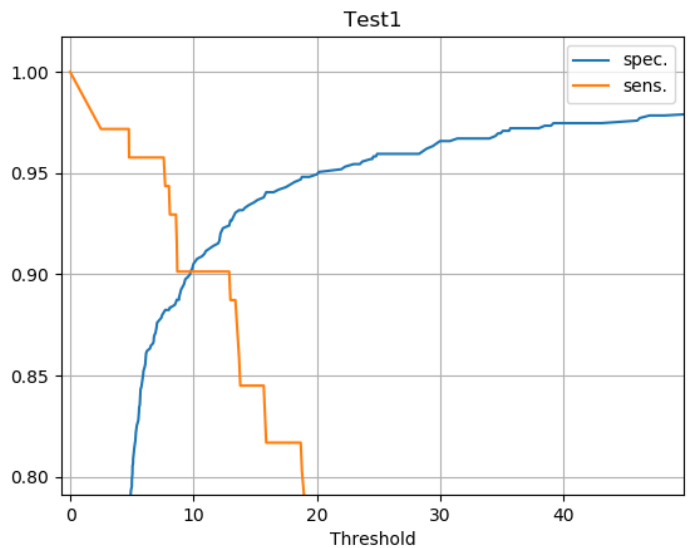
- If the serological test is positive, then the nasopharyngeal swab is used, and after one day or so the true status of the patient is known. Thus **false positives** mean "unnecessary" swab tests, while true positives are simply confirmed by the swab test.
- If the serological test is negative, nothing happens. This means that a false negative case is very dangerous: the person, thinking of being healthy, infects other people and potentially causes their death. Therefore **false negatives** are very dangerous, and the probability of false negatives $P(T_n|D) = 1 - P(T_p|D)$ should be kept as small as possible. This means that sensitivity $P(T_p|D)$ should be as high as possible.

However, it is not acceptable that sensitivity is set equal to 1 by setting the threshold to very low values (the test becomes useless since all the results are positive). It is convenient to plot sensitivity and specificity versus the threshold, to get a clearer picture of the problem.

How to set the threshold [3]

```
16 plt.figure()
17 plt.plot(data_Test1[:,0],1-data_Test1[:,1],label='spec.')
18 plt.plot(data_Test1[:,0],data_Test1[:,2],label='sens.')
19 plt.grid()
20 plt.xlabel('Threshold')
21 plt.title('Test1')
22 plt.legend()
```





How to set the threshold [4]

- When the threshold is zero, sensitivity is 1 but specificity is 0 (not acceptable).
- Specificity and sensitivity are equal (0.9) if the threshold is 10
- A threshold higher than 10 favours specificity, a threshold lower than 8.7 favours sensitivity.
- Since we want to increase sensitivity, we should use a threshold less than 10.

How to set the threshold [5]

Let us assume that COVID-19 prevalence $P(D)$ is 2.5%, and let us evaluate:

- The probability that a person with a negative test is truly healthy:

$$P(H|T_n) = \frac{P(T_n, H)}{P(T_n)} = \frac{P(T_n|H)P(H)}{P(T_n|D)P(D) + P(T_n|H)P(H)}$$

- The probability that a person with a negative test is ill:

$$P(D|T_n) = 1 - P(H|T_n)$$

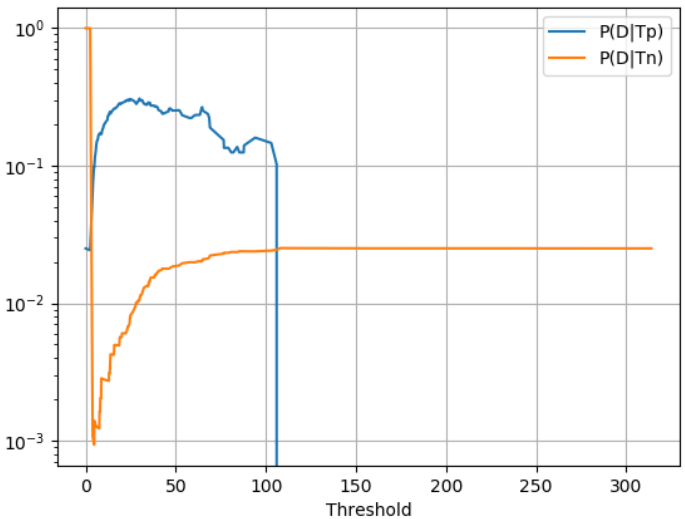
- The probability that a person with a positive test is truly ill:

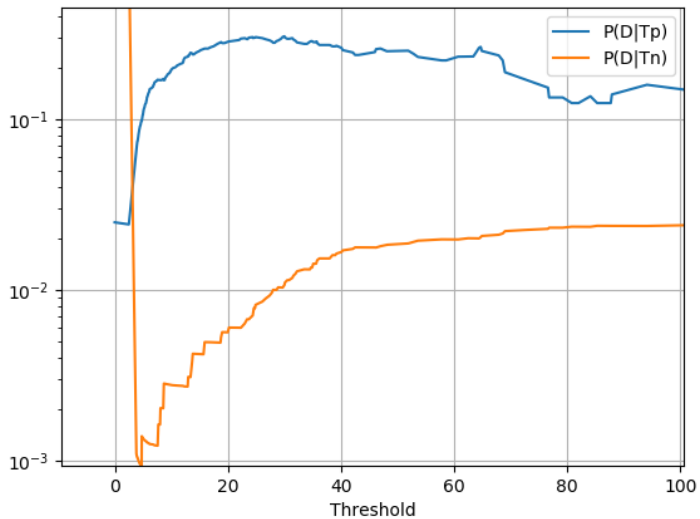
$$P(D|T_p) = \frac{P(T_p, D)}{P(T_p)} = \frac{P(T_p|D)P(D)}{P(T_p|D)P(D) + P(T_p|H)P(H)}$$

How to set the threshold [6]

```
23 prev=0.025
24 spec=1-data_Test1[:,1]
25 sens=data_Test1[:,2]
26 # add 1e-8 to the denominator to prevent division by 0
27 PDgivenTp=sens*prev/(sens*prev+(1-spec)*(1-prev)+1e-8)
28 PHgivenTn=spec*(1-prev)/(spec*(1-prev)+(1-sens)*prev+1e-8)
29 PDgivenTn=1-PHgivenTn
30 plt.figure()
31 plt.semilogy(data_Test1[:,0],PDgivenTp,label='P(D|Tp)')
32 plt.semilogy(data_Test1[:,0],PDgivenTn,label='P(D|Tn)')
33 plt.legend()
34 plt.grid()
35 plt.xlabel('Threshold')
```

How to set the threshold [7]





The minimum value of $P(D|T_n)$ is 10^{-3}, but the corresponding $P(D|T_p)$ is only 0.1 (not acceptable).
If the threshold is set to 7.5, then $P(D|T_n)$ increases to $1.2 \cdot 10^{-3}$ (only a slight increase), but $P(D|T_p)$ becomes 0.17, which is much better.
Therefore 7.5 can be a reasonable threshold that gives:

How to set the threshold [8]

- Sensitivity $P(T_p|D) = 0.96$, false negative probability $P(T_n|D) = 0.04$
- Specificity $P(T_n|H) = 0.88$, false positive probability $P(T_p|H) = 0.12$
- $P(D|T_n) = 1.2 \cdot 10^{-3}$, $P(H|T_n) = 0.9988$
- $P(D|T_p) = 0.17$, $P(H|T_p) = 0.83$.

Table of Contents

What do you have to do?

1. Repeat everything for `Test2`.
2. Complete the report, including a discussion on the characteristics of the two tests, then suggest one and motivate your choice.
3. Report due in 14 days.

Exercise 1 on ID3

Assume X_1, X_2, X_3 binary variables (they can take values 0 or 1). Assume that the class C is built as

$$C = (X_1 \text{OR} X_2) \text{AND} X_3$$

where OR is the logical or and AND is the logical and. Write the table that lists all the possible values of the triplet X_1, X_2, X_3 (8 rows) and the corresponding value of C.
Once you have the table, apply ID3 to build the decision tree. The exercise must be solved with pen and paper.
No report on this, you might be asked about this exercise at the oral exam.

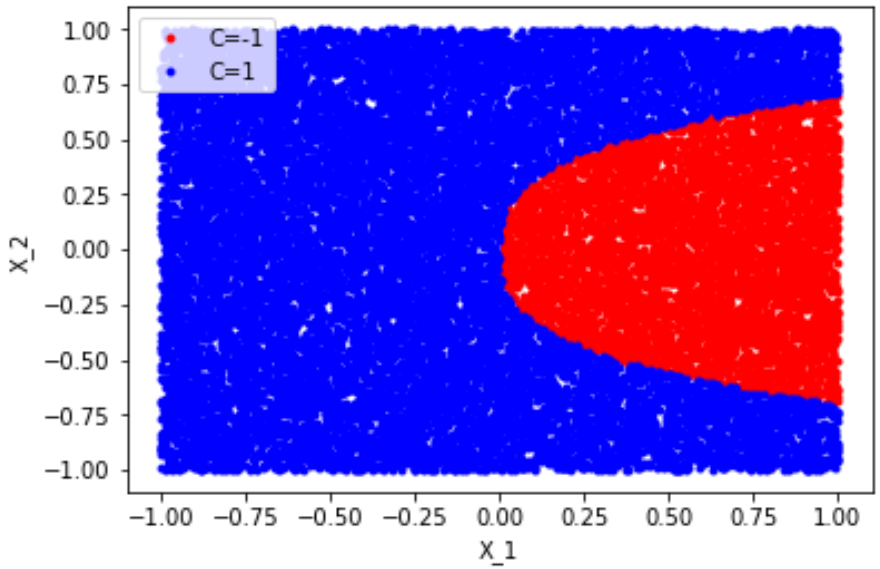
Exercise 2 on C4.5 [1]

- During the lecture we discussed about ID3, which works if all the features are categorical. Algorithm C4.5 is an evolution that allows to manage numerical features. We will discuss C4.5 in the next lecture, but you can start using it as a black box
- The goodness of algorithms is often tested using artificial data (toy problems).

Then consider this toy problem: X_1 and X_2 are two uniformly distributed random variables in the range $[-1, 1]$. Class C is defined as

$$C = \text{sign}\left(-2\,\text{sign}(X_1)\,|X_1|^{2/3} + 4X_2^2\right)$$

Exercise 2 on C4.5 [2]



Write a Python script that

1. Randomly generates N_{train} training points (i.e. X_1, X_2 and the corresponding class). Start with $N_{train} = 100$ and then increase it.
2. Plots X_2 versus X_1 with a red point if the class is -1, with a blue point if the class is +1.
3. Generates the C4.5 decision tree using these training points.

Exercise 2 on C4.5 [3]

4. Tests the decision tree on other $N_{test} = 20000$ randomly taken samples, generating for each of them the estimated class $\hat{C}$.
5. For the test dataset, plots X_2 versus X_1 with a red point if $\hat{C} = -1$, with a blue point if $\hat{C} = 1$.
6. Measures the accuracy, i.e. $P(\hat{C} \neq C)$, on the test dataset.

Python relevant code lines:

```
1 from sklearn import tree
2 clfX = tree.DecisionTreeClassifier('entropy')
3 clfX = clfX.fit(X,C)
4 ...
5 hatCtest = clfX.predict(Xtest)
6 from sklearn.metrics import accuracy_score
7 print('accuracy =', accuracy_score(Ctest, hatCtest))
```

Exercise 2 on C4.5 [4]

To generate the decision tree figure:

```
feat_names=['X_1','X_2']
target_names=['C=-1','C=1']
tree.export_graphviz(clfX,out_file='Tree.dot',feature_names=feat_names,
class_names=target_names, filled=True, rounded=True,
special_characters=True)
```

A file named `Tree.dot` (a simple ACII file) is generated in the working folder. Then, using the shell command

`$ dot -Tpng Tree.dot -o Tree.png`

figure `Tree.png` is generated. If this method does not work, search the web for an alternative solution to plot the decision tree.

No report on this, you might be asked about this exercise at the oral exam.