

ICT for Health

Laboratory # 3

ROC - Covid-19 serological tests

Monica Visintin

Politecnico di Torino



November 25th 2021

Table of Contents

1 Description of laboratory # 3

2 Data analysis

3 ROC

4 Threshold setting

5 What do you have to do?

Description [1]

Covid-19 can be diagnosed using nasopharyngeal swabs or using serological tests (or using lung CT-scans, see lab #4).

Serological tests are faster, but nasopharyngeal swabs are more reliable (higher sensitivity and specificity).

For some subjects 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

- 1 Description of laboratory # 3
- 2 Data analysis**
- 3 ROC
- 4 Threshold setting
- 5 What do you have to do?

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 [1]

The first lines in the script are the following:

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 xx=pd.read_csv("covid_serological_results.csv")
5 swab=xx.COVID_swab_res.values# results from swab: 0= neg., 1 = unclear, 2=pos.
6 Test1=xx.IgG_Test1_titre.values
7 Test2=xx.IgG_Test2_titre.values
8 # remove uncertain swab tests
9 ii=np.argwhere(swab==1).flatten()
10 swab=np.delete(swab, ii)
11 swab=swab//2
12 Test1=np.delete(Test1, ii)
13 Test2=np.delete(Test2, ii)

```

Read the data [2]

If you plot `Test1` values, you see that some values are really very high and isolated.

Use DBSCAN (with appropriately set values of ϵ and M) to remove outliers from `Test1`.

Use method DBSCAN implemented in `sklearn.cluster`.

The code lines are not in the script provided by the instructor, you must add them.

Table of Contents

- 1 Description of laboratory # 3
- 2 Data analysis
- 3 ROC**
- 4 Threshold setting
- 5 What do you have to do?

Find sensitivity [1]

Let us focus on Test2 (**you have to repeat everything for Test1**). We arbitrarily set the threshold to 5, and we find sensitivity and specificity for this threshold.

```
10 x=Test2
11 y=swab
12 x0=x[swab==0] # test results for healthy patients
13 x1=x[swab==1] # test results for ill patients
14 Np=np.sum(swab==1) # number of ill patients
15 Nn=np.sum(swab==0) # number of healthy patients
16 thresh = 5 # example of threshold
17 n1=np.sum(x1>thresh) # number of true positives for the given thresh
18 sens=n1/Np # sensitivity
19 n0=np.sum(x0<thresh) # number of true negatives
20 spec=n0/Nn # specificity
```

Sensitivity is 0.479 and specificity is 0.962.

Find sensitivity [2]

To better understand what is going on, **plot the conditional pdfs of the IgG level given that the swab test is 0, and given the the swab test is positive.** Use `matplotlib.hist` or similar methods, but set `density=True`. This part is not present in the provided script, you must add it.

Find false alarm

ROC (Receiver Operating Curve) is sensitivity $P(T_p|D)$ (y-axis) versus false alarm $P(T_p|H)$ (x-axis):

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

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 0.038. Therefore, when the threshold is set equal to 5, the point in the ROC curve has abscissa 0.038 and ordinate 0.479. We must find sensitivity and false alarm for all the possible thresholds (of course between the minimum and the maximum values of Test2).

ROC curve [1]

Instead of selecting uniformly spaced thresholds between the minimum and the maximum⁴, it is better to consider the thresholds equal to the possible values taken by Test2, from the smallest to the largest, and it is also convenient to write a method/function to get the data from the ROC curve:

```

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)# number of true positives
22         sens=n1/Np
23         n2=np.sum(x0>thresh)# number of false positives
24         FA=n2/Nn
25         data[i,0]=thresh
26         data[i,1]=FA
27         data[i,2]=sens
28     return data
```

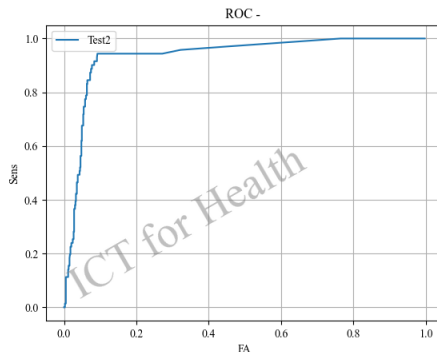
ROC curve [3]

Then in the main part of the script you have

```
1 data_Test2=findROC( Test2 , swab)
2 plt.figure()
3 plt.plot( data_Test2[:,1], data_Test2[:,2], label='Test2' )
4 plt.axis([0,1,0,1])
5 plt.xlabel('FA')
6 plt.ylabel('Sens')
7 plt.grid()
8 plt.legend()
9 plt.title('ROC for Test2')
```

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.

ROC curve [5]

As previously seen, Test1 values range from 2.5 to 314. Thus, the above described method does not consider threshold 0 for Test1, which leads to poor results in that case.

Modify method findROC so that threshold 0 is always present in the output data.

A parameter that is typically used to understand if a test is better than another is to evaluate the area under the ROC curve (AUC): the higher the area, the better the test.

Add your code to method findROC so that it also outputs the area under the ROC curve. Use the rectangular or the trapezoidal rule to evaluate the area.

Note: `sklearn.metrics` methods `roc_curve`, `plot_roc_curve`, `roc_auc_score` cannot be used in this case. The assumption in these methods is that you have the output of a classifier that provides the estimated probability of class 1 and the true class (1 or 0), which you do not have (we are not classifying anything).

Table of Contents

1 Description of laboratory # 3

2 Data analysis

3 ROC

4 Threshold setting

5 What do you have to do?

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 (less for the rapid swab test) 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.

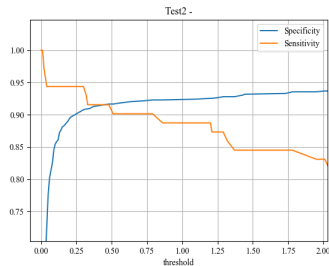
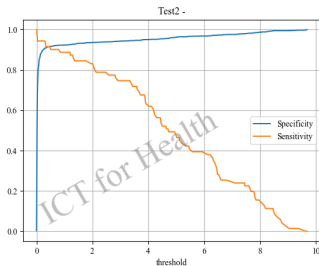
However, it is not acceptable that sensitivity is set equal to 1 by setting the threshold to very low values (the test becomes useless since all the results are positive). It is convenient to plot sensitivity and specificity versus the threshold, to get a clearer picture of the problem.

How to set the threshold [3]

```

16 plt.figure()
17 plt.plot(data_Test2[:,0],1-data_Test2[:,1],label='spec.')
18 plt.plot(data_Test2[:,0],data_Test2[:,2],label='sens.')
19 plt.grid()
20 plt.xlabel('Threshold')
21 plt.title('Test2')
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 (0.92) if the threshold is 0.45
- A threshold higher than 0.45 increases specificity, a threshold lower than 0.45 increases sensitivity.
- Since we want to increase sensitivity, we should use a threshold less than 0.45.

How to set the threshold [5]

Let us assume that COVID-19 prevalence $P(D)$ is 2%, 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:

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

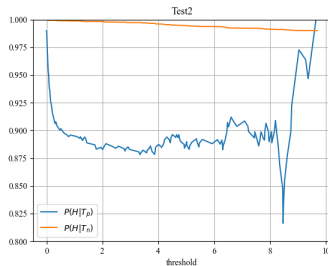
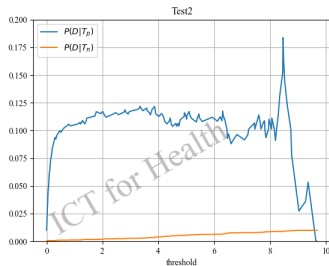
How to set the threshold [6]

Add the Python code that allow you to plot:

- $P(D|T_n)$ and $P(D|T_p)$ versus threshold in one figure (to ease comparison)
- $P(H|T_n)$ and $P(H|T_p)$ versus threshold in another figure (to ease comparison)

How to set the threshold [7]

Note: the figures in this slide were obtained with different parameters.



What is a reasonable threshold, then?

Table of Contents

1 Description of laboratory # 3

2 Data analysis

3 ROC

4 Threshold setting

5 What do you have to do?

What do you have to do?

- 1 Complete the assignments given in the previous slides.
- 2 Repeat everything for Test1.
- 3 Complete the report.
- 4 In the conclusions compare the results obtained for the two tests and suggest which to use, if you believe that one is better than the other.
- 5 Report due in 14 days (Wednesday December 8th 2021, 11.59 PM).
Maximum number of pages: 5.

Next labs

Next labs will be on

- Decmeber 9th (lab #4), report due on Wednesday December 22nd at 11.59 PM.
- December 16th (lab #5), report due on Thursday December 30th at 11.59 PM.

There will be one more lab in January, entirely managed by the instructor, on EEG (no report).