Image and signal processing: Computer exercise 1

CE01G02T05

PRULIERE Valentin SID-LAKHDAR Riyane

26/09/2015

Abstract

This report sumarise, explains and refers to the answers we have designed for the first Ïmage and Signal Processingc̈omputer exercise.
Linked to this report you will find s̈rc/d̈irectory containing the Python source code corresponding to the exercise' expecations. You will find eather an ötput/d̈irectory containing some output given by our programs.

1 Introduction

The aim of the current computer exercise is to have a first approch with a set of basis theoretical knowledge conserning signal processing.
It eather leads to implement a set of basis programs to create, observe and analyze signal behaviours.
Thus, we have tried to separate as much as possible our theoretical reseaches from the output results. We have eather tried to implement our programs regardingless to the expected theoretical expectations.
In this purposes, this rapport design tries to respect the convention carried on the given rapport examples. Thus, the two main sections of this rapport:

- M̈aterial & Methods:̈ In this section, we described the theorical methodes used to answer to every question. We provide eather the expected algorithms, them main principles, purposes and eventually them implementation in python.
- Ëxperiments:̈ In this section, we give the experimental result of our experiences. We try to explain them, and see whether they are consistant whith the theoretical expecations or not.

Furthermore, to try to make our implementations and results as user frindly as possible the programatic convention we have used is:

- To each python file src/xxx.py corresponds a text file output/xx.txt where the main result of the xx program are stored.
- The program names are explicitly linked to the name of the corresponding question, or to the name expected by the question.
- The main algorithms which have been implementd are given in the M̈aterial & Methods; and them results are given and described in the Ëxperimentss̈ection.

2 Material & Methods

2.1 Exercise 3: Financial Experts

2.1.1 Question 3.1

To create the signal corresponding to the experts prediction, we implemented the file experts.py where we:

- Create the signal using its sample values (np.array)
- Plot the signal using the same methodes as in the previous exercise

Our algorithme is given by the following program experts.py

```
1   import numpy as np
2   import matplotlib.pyplot as plt
3
4
5
6   # Definition of expertID and expertPrediction tables
7   expertID = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
8   expertPrediction= np.array([996, 868, 855, 956, 867, 933, 866, 887, 936,
        901, 818, 956])
9
10  plt.figure()
11  line = plt.plot(expertID, expertPrediction, '-', marker='h', linewidth=2)
12
13  plt.grid()
14  plt.title('Financial predictions...')
15  plt.ylabel('Stock market price prediction')
16  plt.xlabel('Financial expert')
17  plt.show()
18
19  # Saving the signal
20  # Please refer to http://docs.scipy.org/doc/numpy/reference/generated/
        numpy.savetxt.html
21  np.savetxt('../output/experts.txt', (expertID, expertPrediction))
```

2.2 Exercise 4: Mean and Standard Deviation

2.2.1 Question 4.2 & 4.3

The mean and the standard deviation function has been implemented in the following Python program.

```
1   def myMean(signal):
2     res = signal[0]
3     for k in range(1, len(signal)):
4             res = res + signal[k]
5     return res/len(signal)
6
7
8   def myStandardDeviation(signal):
9     mean = myMean(signal)
10    s = 0
11    for k in range(len(signal)):
12            s += math.pow((signal[k] - mean),  2)
13    return math.sqrt(s / (len(signal)-1))
```

Figure 1: Python code to compute the mean and the standard deviation of a signal.

2.2.2 Question 4.4

Laying on the previous algorithms, we can easily compute the mean and the standard deviations of the previous experts signal.¡/br¿ The corresponding program is given in the program ispFunctions.py

```
from experts import expertPrediction, trueValue


mean         = myMean(expertPrediction);
deviation    = myStandardDeviation(expertPrediction)
f            = open("../output/ispFunction.txt", 'w')
f.write("Mean of the experts prediction: " + str(mean))
f.write("Standard deviation of the experts prediction: "+ str(deviation))
```

2.2.3 Question 4.5

Using the previous programs and the definition of the accuracy, we have implemented the next algorithm to compute the accuracy of a signal. The two last ligns of the program provide the way to use this algorithm to compute the accuracy of the expert's signal.

```
    # tv: true average value of the signal (unlike the mean which comes
        from experimental values)
    def myAccuracy(signal, tv):
      mean = myMean(signal)
      return (math.fabs(tv - mean))

    accuracy = myAccuracy(expertPrediction, trueValue)
    f.write("Accuracy of the experts prediction: " + str(accuracy))
```

2.3 Exercise 5: Histogram

2.3.1 Question 5.1

Given a signal s:

$$s = \{x_0, ..., x_{N1}\} \tag{1}$$

, and a value i that bellongs to [0, N-1]. We know that:

$$x_i \text{ counted in } h_j \Leftrightarrow \tag{2}$$

$$minVal + j * l \leq x_i < minVal + (j+1) * l \Leftrightarrow \tag{3}$$

$$j * l \leq x_i - minVal < (j+1) * l \Leftrightarrow \tag{4}$$

$$\text{assuming that } l \neq 0 : j \leq \frac{x_i - minVal}{l} < j + 1 \Leftrightarrow \tag{5}$$

Thus, for a given i, $x_i$will be counted in h_j if and only if $j = int(\frac{x_i - minVal}{l})$

$$\tag{6}$$

2.3.2 Question 5.2

According to the general definition of a histogram, and to the specific purposes described in this question, we can say that the expected histograms of this exercise must show the number of signal samples belonging to each range of value. Thus, given a size of values range, our algorithme, will compute the number of diffrent ranges of our histogram (amplitude of the historgam). Then, it will use the previous property to compute the number of signal samples that bellong to this range. The signal samples are gone through a single time.

```
1   def myHistogram(signal,intervalLength):
2       minVal = myMin(signal)
3       maxVal = myMax(signal)
4       length = len(signal)
5       N = int(ceil((maxVal-minVal)/intervalLength)) + 1
6       h = [0]*N
7       histoAxis = [m * intervalLength + minVal for m in xrange(N)]
8       for i in range(0,length - 1):
9               h[int(ceil((signal[i] - minVal)/intervalLength))] += 1
10
11      return h,histoAxis
```

2.4 Exercise 6: Random signal

2.4.1 Question 6.1

Guiven a random variable X and its corresponding probability function p, we have:

$$E(X) = \int_{-\infty}^{\infty} x * p(x)\,\mathrm{d}x \tag{7}$$

$$= \int_{-\infty}^{a} x*p(x)\,\mathrm{d}x + \int_{a}^{b} x*p(x)\,\mathrm{d}x + \int_{b}^{-\infty} x*p(x)\,\mathrm{d}x \text{according to the Chasles' Theorem} \tag{8}$$

We know that the possible values of the variable X bellongs to [a, b]. So p(x) = 0 for each x out of [a, b]. Thus:

$$E(X) = 0 + \int_{a}^{b} x * p(x)\,\mathrm{d}x + 0 \tag{9}$$

Furthermore, as X is a **uniformly** random variable, the p function does not depend on its entry x. Otherwise, the probability that X equals a fixed element from [a, b] is given by the formula:

$$\frac{numberoffavorablecases}{numberofpossiblecase} = \frac{1}{b-a} \text{(continuous distribution)} \tag{10}$$

Thus, assuming that b and a are diffrents:

$$E(x) = \frac{1}{b-a} * \int_{a}^{b} x * p(x)\,\mathrm{d}x \tag{11}$$

$$= \frac{1}{b-a} * \frac{b^2 - a^2}{2} \tag{12}$$

$$= \frac{b+a}{2} \text{which proves the expected property} \tag{13}$$

2.4.2 Question 6.2

To prove the given formula, we will first consider expectation of X square.

$$E(X^2) = \int_a^b \frac{1}{b-a} x^2 dx \text{ (using the same arguments as in the previous question)} \tag{14}$$

$$= \frac{1}{b-a} \int_a^b x^2 dx \tag{15}$$

$$= \frac{1}{3 * (b-a)} (b^3 - a^3) \tag{16}$$

$$= \frac{b^2 - ab + a^2}{3} \tag{17}$$

thus:

$$\sqrt{E(X)^2 - E(X^2)} = \sqrt{\frac{b^2 - ab + a^2}{3} - \frac{(a+b)^2}{4}} \tag{18}$$

$$= \sqrt{\frac{a^2 - 2ab + b^2}{12}} \tag{19}$$

$$= \sqrt{\frac{(a-b)^2}{12}} \tag{20}$$

thus

$$\sigma_\chi = \frac{(a-b)}{\sqrt{12}} \tag{21}$$

2.4.3 Question 6.3

Within this question, we wanted to observe the evolution of the distance between the thorical and the observed values of the mean and the standard deviation. Thus we have implemented a program which generates sevral random signals given by them samples. Each sample size is unique. Then, for each signal we can:

- Compute the theorical value of the mean and the standard deviation. This value is given by the previous questions and only depends on the abscissa bounds of the signal.
- Compute the measured mean and myStandardDeviation of the signal using the function previously implemented in ispFunctions.py

Then, for each signal, we keep the ratio of the two mean values, and the ration of the two myStandardDeviation values.
This algorithm has been implemented in the ispFunction.py file and is shown in the next figure:

```
1 #question 6
2 randomSignalSize= np.array([10000, 20000, 30000, 40000, 50000, 60000, 70000,
      80000, 90000, 100000])
3 meanDistance     = np.array(range(0, size(randomSignalSize)), dtype=float)
4 stdDeviationDistance= np.array(range(0, size(randomSignalSize)), dtype=float)
5 for i in range(0, size(randomSignalSize)):
6     signal       = np.random.random(randomSignalSize[i])
7     mean         = myMean(signal)
8     deviation    = myStandardDeviation(signal)
```

```
9       a           = 0
10      b           = randomSignalSize[i]-1
11      meanDistance[i]             = 10*mean       / ((a+b)/2)
12      stdDeviationDistance[i]     = deviation     / ((b-a)/math.sqrt(12))
13
14
15  plt.figure()
16  line = plt.plot(randomSignalSize, meanDistance, '--', color='blue', linewidth
        =2)
17  line = plt.plot(randomSignalSize, stdDeviationDistance, '-', color='red',
        linewidth=2)
18
19  plt.grid()
20  plt.title('Theoretical VS computed values...')
21  plt.xlabel('Number of sample by signal')
22  plt.ylabel('Ratio of the theoretical and the computed value')
23  plt.show()
```

In the 10th line of the algorithm, we can see that the distance between theorical and measured value of the mean is multipyed by 10 for each signal. This 10 coefficient is used to make the corresponding plot more user frindly. Thus, the corresponding plot will bee in the same values range as the standard deviation distance plot. This coefficient does not change the increasing or decreasing character of the distance function.

2.4.4 Question 6.4

The probability distribution function of a random variable X bellonging to [a, b] is known as:

$$\chi : [a, b] \to [0, 1] \tag{22}$$

$$x \mapsto p(\chi < x) : \text{the probability that X be lower than x} \tag{23}$$

Thus, for an equiprobable random variable X,

$$p(\chi < x) = \frac{x - a}{b - a} \tag{24}$$

Secondly, we know by the definition that the probability density function of a random variable is its derivte over x (x represents its single variable).
So the probability density function F of an equiprobable random variable is:

$$F(x) = \frac{1}{b - a} \tag{25}$$

Assuming that a = 0, and b = 1, we have

$$F(x) = 1 \tag{26}$$

3 Experiments

3.1 Exercise 2: Signal Storage and Display Example

3.1.1 Question 2.1

The given program gives a partial definition of a signal (given by a sample of some of its values). It eather shows how to plot it using simple python graphical librarys. The output plot of this program is given by the figure 2:

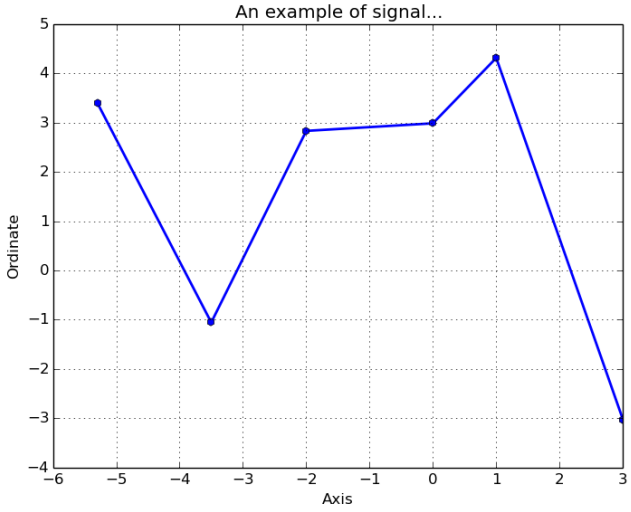

Figure 2: Example of a sample signal plot realised in a python

3.2 Exercise 3: Financial Experts

3.2.1 Question 3.1

Our program (desribed in the material and methods sections) allowed us to plot the signal corresponding to the expert predictions. This output plot is given by the figure: 3.

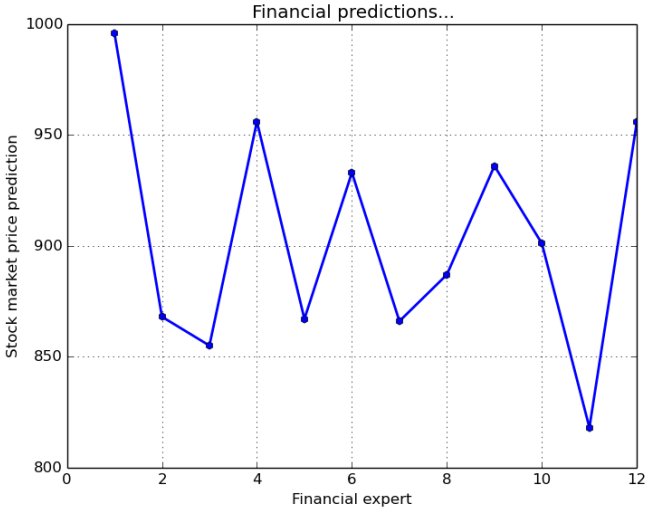

Figure 3: Expert's signal plot

3.3 Exercise 4: Mean and Standard Deviation

3.3.1 Question 4.4 & 4.5

Using the program described in the M̈aterial & Methodss̈ection, we have computed the following values for the experts signal:

- Mean of the experts prediction: 903
- Standard deviation of the experts prediction: 52.1666908704
- Accuracy of the experts prediction: 27.0

3.4 Exercise 6: Random signal

3.4.1 Question 6.3

The figur 4 has been obtained by runing the algorithm described in the algorithm of the section 2.4.3.
The figure represents the evolution over the signal samples size of the ration of:

- Theoretical and measured values of the mean of the signal: blue broken curve
- Theoretical and measured values of the standard deviation of the signal: red continue curve

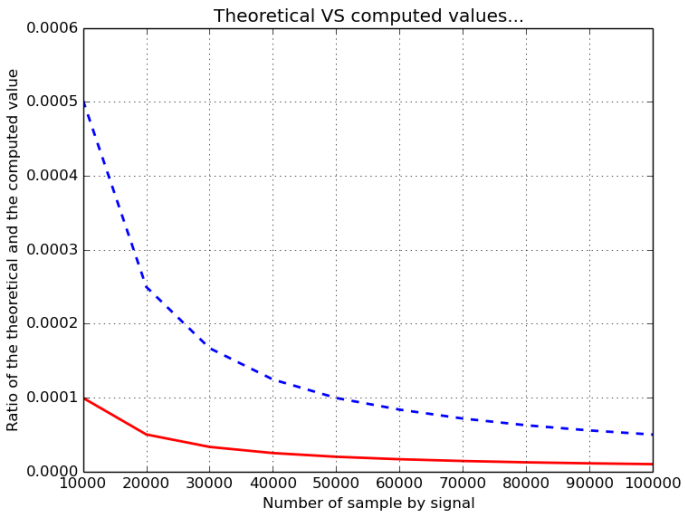

Figure 4: Evolution of the ration of the thoretical and measured values of the mean and the standard deviation

This plot illustrates the fact that the mor samples we generate, the closer the measured mean and standard deviation get to the theoretical mathematical expectation and standard deviation of the underlying random variable.

4 Conclusion

This current rapport trys to summerise the experiments we have been dealing with durring this first computer exercise. It analyzes the experimental results and explains whether they sute or not the theoretical expectations.
During this practical sessions, we have eather ben able to develope some functionality for signal visualization and processing. However the developed algorithms seem verry simple, they may already give an intresting vision about some signal behaviours.