# Master in Big Data Analytics

Academic Year 2016-2017

## Introduction to *Numpy* and *Bokeh*

*Task 2*

*Solving Systems of Equations by using Basic Lineal Algebra functions of* Numpy

Jon Ander Gómez Adrián
jon@dsic.upv.es
Departament de Sistemes Informàtics i Computació
Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

## Contents

**1 Motivation and Objectives** **1**

**2 Tasks** **2**

**3 Conclusions** **4**

**4 Bibliography** **4**

# 1 Motivation and Objectives

The purpose of this task is to help students to catch the ideas explained during classes and put into practice the use of basic and common functions of *Numpy* and *Bokeh*.

The fundamental aspect we are interested in transmitting to students is that common operations of Basic Linear Algebra are already implemented in existing libraries. We should use the most suitable library for our development instead of coding the algorithms again by ourselves.

# 2 Tasks

In order to evaluate the advantages of using *Numpy* matrix operations against the task of coding well-known algorithms already implemented and available in libraries. Students should use the functions provided in different modules of *Numpy*, as it is the module `linalg`. Students should also work with vectors and matrices directly by using the provided functions.

The goal of this exercise is to foster in students the habit of searching in the documentation of *Numpy* in order to find the proper operation.

Additionally, students can see simple examples for reading/writing data from/to files. The Python code for these I/O operations is provided. Students don't have to start from scratch.

In this task a system of equations is provided in a file as follows:

```
5
    18     7    16     8    15  1016
    19    10    11     1     0   652
    18    17     4    18     6   857
     0    12     3     4    11   498
     8    15     4     3    10   654
SOLUTION:     14    17    19     7    19
```

Where the first line contains $n$, the number of equations and unknowns, which is also the number of rows and columns in matrix $A$ and the number of rows in vector $b$. The next $n$ lines contain each one a row of matrix $A$ and one element of $b$. Finally, the solution is provided for checking whether the obtained result matches the correct one.

We look for the solution to $A \cdot x = b$, that expanded to matrices when $n = 3$ with the starting index equal to 0 is:

$$\begin{pmatrix} a_{0,0} \cdot x_0 & a_{0,1} \cdot x_1 & a_{0,2} \cdot x_2 \\ a_{1,0} \cdot x_0 & a_{1,1} \cdot x_1 & a_{1,2} \cdot x_2 \\ a_{2,0} \cdot x_0 & a_{2,1} \cdot x_1 & a_{2,2} \cdot x_2 \end{pmatrix} = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,0} & a_{2,1} & a_{2,2} \end{pmatrix} * \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

The following code generates a random system of equations that ensures all unknowns are integer numbers.

```

import numpy

n=5

A = numpy.random.rand( n, n ) * 20
```

```
A = numpy.floor( A )
A = A.astype( int )

print( A )

x = numpy.random.rand( n ) * 20
x = numpy.floor( x )
x = x.astype( int )

print( x )

b = numpy.dot(A,x)

print( b )

f=open( 'system.txt', 'w' )
f.write( "%d\n" % n )
for i in range(n):
    for j in range(n):
        f.write( " %5d" % A[i,j] )
    f.write( " %5d\n" % b[i] )

f.write( "SOLUTION: " )
for i in range(n):
    f.write( " %5d" % x[i] )
f.write( "\n" )
f.close()
```

The following code should be used for loading a system of equations from a file on disk.

```

import numpy


def load_equation_system( filename ):
    f=open(filename, 'r' )
    # Reads 'n', the number of equations
    line=f.readline()
    tokens=line.split()
    n = int(tokens[0])
    # Creates the Numpy arrays for A and b
    A = numpy.zeros( [n,n] )
    b = numpy.zeros( n )
```

```
14      for row in range(n):
15          line=f.readline()
16          tokens=line.split()
17          for column in range(n):
18              A[row,column] = int(tokens[column])
19          b[row] = int(tokens[-1])
20      f.close()
21
22      return A,b
```

Students should write the Python code for:

1. Loading $A$ and $b$ from a file on disk.

   The file can be generated by running the Python program `equation-system-generator.py`

2. Solving the equation, i.e. obtaining the values for $x_i$

3. Show the results in order to check if they match with the solution provided.

Students should deliver to teacher a report, with no more than six A4 pages, explaining the different tasks carried out and including the Python code implemented in this task with the proper comments. This will be evaluated to contribute to the grade of the subject this class belongs to.

**Estimated time 2 hours.**

# 3 Conclusions

As said in the Introduction, the purpose is to introduce students in the use of *Numpy* for writing efficient Python programs and taking advantage of *Bokeh* functionality in order to represent different kinds of data: data to be analyzed and results.

# 4 Bibliography


- `http://docs.scipy.org/doc/`
- `http://docs.scipy.org/doc/numpy/user/`
- `http://wiki.scipy.org/Tentative_NumPy_Tutorial`