Master in Big Data Analytics

Academic Year 2016-2017

Introduction to ***Numpy*** and ***Bokeh***

Task 1

Comparison of running times between pure Python code and code using 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 **3**

4 Bibliography **3**

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 operations involving vectors and matrices are several orders of magnitude faster when using *Numpy* functions instead of write Python code for accessing this data structures.

Additionally, *Numpy* offers efficient implementations of basic linear algebra operations for matrices.

In parallel, we are also interested in introducing the *Bokeh* library to students (`http://bokeh.pydata.org`). *Bokeh* is a set of modules for easily plotting different kinds of data. We will focus our attention into some modules of *Bokeh*: `plotting`, `palettes`, `charts` and `sampledata`.

2 Tasks

In order to evaluate the advantages of using *Numpy* matrix operations against the use of loops in Python, students should write several programs for performing the matrix operations described below.

The goal of this exercise is to measure running times of two or three alternative codes in Python, one taking advantages of *Numpy* matrix operations and the other ones using Python loops.

For the different example programs students should save the running times for different input sizes in an array or a list, then use it to show the evolution of running times of the two alternatives as input size of the problem grows.

1. Sorting a list (provided as an example: `task1.py`)

   (a) By using Python loops.

   (b) By using one Python loop and argmin function.

   (c) By using the sort algorithm provided as a function.

2. Scalar product of two arrays:

   (a) By using lists and Python loops.

   (b) By using *Numpy* arrays and Python loops.

   (c) By using *Numpy* arrays and matrix operations.

3. Vector-Matrix product and/or Matrix-Vector product:

   (a) By using lists and Python loops.

   (b) By using *Numpy* arrays and Python loops.

   (c) By using *Numpy* arrays and matrix operations.

4. Matrix product:

   (a) By using lists and Python loops.

(b) By using *Numpy* arrays and Python loops.

(c) By using *Numpy* arrays and matrix operations.

Students should complete the Python code making use of *Numpy* and *Bokeh*, taking as a reference the code in `task1.py`, then representing the evolution and explaining the different tasks carried out.

All must be part of a report with no more than 10 A4 pages, that will be evaluated to contribute to the grade of the subject this class belongs to.

Estimated time 3 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`
- `http://wiki.scipy.org/Numpy_Example_List`
- `http://bokeh.pydata.org/`
- `http://pandas.pydata.org/`
- Another library for visualizing data is `http://matplotlib.org/`