

Benchmarking Octave, R and Python platforms for code prototyping in Data Analytics and Machine Learning applications programming

Abstract

Octave, R and Python identical codes are tested in terms of in terms of end-user execution speed, using a very low-end "embedded" hardware system and a standard office workstation. The codes include algorithmic primitives common in Data Analytics and Machine Learning, i.e., matrix manipulation (inversion, product), linear Algebra, linear regression, Singular Value Decomposition (SVD), fast Fourier transformation (FFT) and a baseline Bubblesort implementation for testing flow control structures.

Description

In Data Analytics and Machine Learning, code prototyping is an integral part of the Research & Development (R&D) process, especially in data exploration and algorithm design. The programming tools and platforms used for these tasks are selected for rich API/library base, high-level expression syntax, very compact code, interactive on-the-fly code input, abstract data management and best-possible execution speed. Thus, traditional programming languages are usually inappropriate for such heavily iterative and exploratory coding evolutions.

Today, by far the three most popular and appropriate choices are Octave, R and Python. In this work, these three programming environments are assessed in terms of end-user execution speed. More specifically, some common algorithmic primitives are implemented and tested in each language separately, including matrix manipulation (inversion, product), linear Algebra, linear regression, Singular Value Decomposition (SVD), as well as fast Fourier transformation (FFT) as a standard procedure in a signal processing pipeline. Additionally, a baseline implementation of the Bubblesort algorithm is employed for testing the efficiency of flow control structures and execution performance in code branching.

The results present the performance of the three identical source codes in terms of end-user execution speed (elapsed time) in three different hardware platforms, namely: (1) simulating very low-end processing and resources machine similar to embedded systems (Linux, 2GB RAM, N20 Atom single-core CPU), (2) a standard/enhanced office workstation (Win10, 16GB RAM, dual-core i7 CPU) and (3) a high-end workstation or small office server (Win10, 32GB RAM, quad-core i7 CPU).

Note for the session: Less than 30 minutes in total, digital projector with hdmi/vga input (oral presentation).

Selected track: "Science & Education"