Big Data and Security

Jeffrey Borowitz, PhD

Lecturer

Sam Nunn School of International Affairs

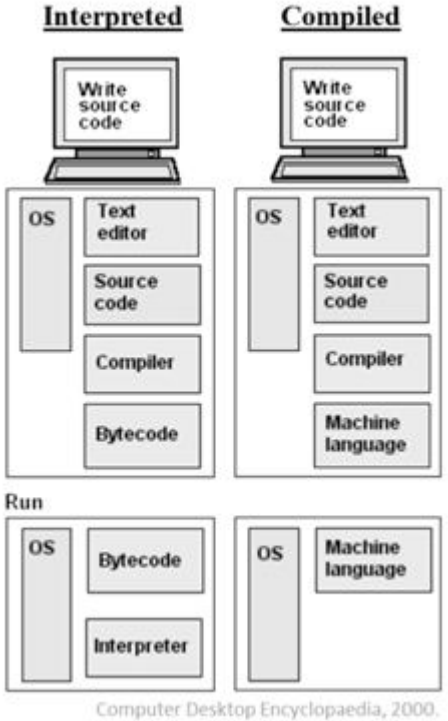
Programming Languages and Tradeoffs

Programming Languages: Tradeoffs

- Think about what a computer is doing:
  - Loads some stored data
  - Performs some computations on stored data
  - Writes results back somewhere
- If you micromanage every piece of this process, you can make the computer do a given task optimally
- In practice, this is very hard - we think in abstractions
  - Low level: move bits to CPU register
  - High level: for loops, finding an element in a list

Levels of Programming Languages

- We write source code
- Sometimes it is interpreted
  - A program looks at the commands and then translates that into machine code
  - This is how e.g. Python and R (which we will use) work
  - This facilitates rapid, interactive work
  - But it's slower than if the program could be compiled
- Sometimes it is compiled
  - Compiling takes a long time
  - But you can search for more ways to optimize your computation on the CPU
  - So programs in these languages are faster



Source: https://github.com/carlosnasillo/Hybrid-Genetic-Algorithm/blob/master/README.markdown

Georgia Tech

Programming Language Performance

	Fortran gcc 4.8.2	Julia 0.3.7	Python 2.7.9	R 3.1.3	Matlab R2014a	Octave 3.8.1	Mathe-matica 10.0	JavaScript V8 3.14.5.9	Go go1.2.1	LuaJIT gsl-shell 2.3.1	Java 1.7.0_75
fib	0.57	2.14	95.45	528.85	4258.12	9211.59	166.64	3.68	2.20	2.02	0.96
parse_int	4.67	1.57	20.48	54.30	1525.88	7568.38	17.70	2.29	3.78	6.09	5.43
quicksort	1.10	1.21	46.70	248.28	55.87	1532.54	48.47	2.91	1.09	2.00	1.65
mandel	0.87	0.87	18.83	58.97	60.09	393.91	6.12	1.86	1.17	0.71	0.68
pi_sum	0.83	1.00	21.07	14.45	1.28	260.28	1.27	2.15	1.23	1.00	1.00
rand_mat_stat	0.99	1.74	22.29	16.88	9.82	30.44	6.20	2.81	8.23	3.71	4.01
rand_mat_mul	4.05	1.09	1.08	1.63	1.12	1.06	1.13	14.58	8.45	1.23	2.35

- C is faster because it is compiled
- R and Python are slower because they are interpreted
- R and Python both call C functions for the most processing intensive functions

Source: https://cs.stackexchange.com/questions/45486/how-can-a-language-whose-compiler-is-written-in-c-ever-be-faster-than-c

Georgia Tech

So Why Would You Learn R/Python?

- Community also matters!
  - Python and R have strong communities related to data analysis
  - Most new statistical/machine learning methods are first published in R, then in Python, and never in C
  - There are big enough communities that e.g. people have written interfaces with lots of other useful programs
- And some languages are built more for some things than others.
  - A lot of what takes time in data analysis is actually more like the rand mat mul, where
  - Python and R are only 3x slower than C instead of 30 or 400x slower

Python

- It's easy to learn (compared to other stuff!!!)
- It has an elegant syntax (compared to other stuff!!!)
- It has nice tools for data analysis
  - NLTK - Natural Language Toolkit
  - SciPy - a collection of programs implementing
    - Numpy - Low level numerical computation in
    - Python Pandas - Higher level data analysis tools in Python
    - Scikits - scikit-learn (http://scikit-learn.org/stable/) has a lot of machine learning algorithms
    - Matplotlib - Plotting
    - Jupyter Notebook - a more analysis-centric way of organizing code and outputs than scripts

R

- R is a little uglier than Python
- But it has the most full featured statistical libraries
- It has a bigger data analysis community than Python
- R makes it easier to do basic data analysis than Python

Lesson Summary

- Programming languages are either compiled (ex. C) or interpreted (ex. Python and R)
- Interpreted languages like Python and R are slower than C, but are still useful due to their large communities and ability to execute the tasks they were built for well
- Python is relatively easy to learn and has nice tools for easy analysis
- R has the most full-featured statistical libraries and is easier to use for statistical analysis than Python