BIG DATA: AN OVERVIEW

What counts as big data?

- Any amount of data that breaks down our typical processes is considered big data
- Rule of thumb: if you can fit the data source in a high-end computer's RAM, it's not big.
- Why does this need a special name... it's just data?

Four Major Issues - Volume

Volume of data

- Relational Databases become a bottleneck at large scale
- Vertical scaling (getting a bigger hard drive) is massively expensive
- Horizontal scaling (binding many hard drives together) is massively complicated

Four Major Issues - Velocity

Speed of processing

- Most data science processes are at least O(n)
- If you parallelize your work, how do you make sure no date is left behind?
- How do you parallelize on a dataset that's 60TB?

Four Major Issues - Variety

So many data types

- We can design a system to handle DataFrame like data
- But we also want systems that can handle images, text, sound, categories, etc, etc
- Our data may not be static: can we handle streaming data?

Four Major Issues - Veracity

Data quality is way more challenging

- If you have 1000 samples, having 4 standard deviation process occur is extremely rare.
- If you have 1,000,000,000 samples, a 4 standard deviation process will occur nearly 1M times.
- When you record more data, there are more chances for your data to be weird.

MORAL OF THE STORY: WE NEED TOOLS TO HANDLE BIG DATA

Three Approaches

- Out-of-Core processing using Dask
- MapReduce and Hadoop (the grandfather of big data)
- Spark

Rules of Thumb

- If it fits in RAM, just use Pandas/numpy
- If it doesn't fit in RAM, but can still be processed on your computer (20-50ish GB), use Dask locally
- If it's bigger than that, use Spark or Dask as a cluster. See here for a discussion: http://docs.dask.org/en/latest/spark.html

QUESTIONS?