Big Data in the Real World

Alex Haigh
CS102 Winter 2019
A reminder...

“Big Data” is about the extraction of actionable or useful information from very large datasets
But, to do anything useful...

Data needs to be

- Collected and Cleaned
- Stored and Managed
- Analyzed

These all change when you have gigantic datasets!
We’ve spent all of this class looking at small data...

It fits in your computer’s memory and can be read (entirely) with 2 lines of code!

Some reference points:

- Computer Hard Drive = 250 gb
- Computer RAM = 8 gb
- Library of Congress = 10,000 gb

What about the data we worked with?

- allData.tsv = 0.0018 gb (~5-10 thousand movies)
- Wines10k.csv = 0.003 gb
- Chicago Crimes = ~5 gb
But we generate data at a massive scale!

The Netflix Prize dataset had 100 million ratings of 17,770 movies from 480,189 customers.

Google has 8 services with over 1 billion users!

Amazon has 600+ million products.

The SLAC LSS Telescope will generate 15 tb of data every night for 10 years.
Detour on smart speakers

There are 120 million smart speakers in circulation in the US (source)

Every hour of audio is ~0.005 gb

They collect 600,000 gb of audio every hour

That’s 60 libraries of Congress!
Data collection and cleaning

It’s more complicated than `pd.load_csv`

Data is dirty. In particular, it’s often:

- Irregular, missing, incomplete
- Unstructured
- Incorrect (human error)
- Spread out across different databases and files

Even clean data can be challenging. In many applications, data is:

- Dynamic and constantly changing
- Urgent: you need to respond to a query in real-time
Possible consequences (this is real life!)

What happens if you write a ML classifier that depends on feature X, and you try to classify a point that is missing a value for X?

How would you mine Electronic Medical Records that are unstructured, not standardized, and often missing data?

How would you tie together data from different sources - what if they are in different databases, on different computers, or in different datacenters (different continents!)?

How do you efficiently query databases with billions of entries?
Ethics of data collection

When is it *ethical* to collect data?

- When should an app collect your location?
- How should Apple collect and store the heart rate data from my Apple Watch?
- When do I need to get your permission before collecting data?

When is it *legal* to collect data?

- **COPPA** = Children’s Online Privacy Protection Act
- **HIPAA** = Health Insurance Portability and Accountability Act
- **GDPR** = European Union General Data Protection Regulation
Storing and managing data at scale

DBMS is a $65bn annual market

Why are databases good at scale?

- SQL
- Query optimization
- Transactions
  - Manage concurrency, recovering from crashes

Challenges

- Replication - sometimes have to manually partition your data
- Security breaches
Security breaches *happen all the time*

All of these companies have been hacked in the last 5 years!
SQL Injections

Hi, this is your son's school. We're having some computer trouble.

Oh, dear - did he break something? In a way...

Did you really name your son Robert'); DROP TABLE Students;-- ?

Oh, yes. Little Bobby Tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.

https://xkcd.com/327/
How do you analyze data at scale?

*Simple techniques work very well when you have a lot of data*

Do the computation in parallel!

- MapReduce paradigm and Spark

Computing hardware and software optimized to run these algorithms

- Programming Languages for AI: PyTorch, Tensorflow
- GPUs, TPUs - multiple billion dollar companies out of this
What is MapReduce?

MapReduce is a programming paradigm that automatically:

- Runs programs in parallel
- Manages hardware/software failures (e.g. a server crashes)
- Processes very-large-scale data

Invented at Google in 2003 by Sanjay Ghemawat and Jeff Dean

Made it manageable for ordinary coders to write distributed programs

Hadoop (open-source implementation of MapReduce), Spark (extension of MapReduce), and other advanced variants are the primary tools used today to write distributed programs.
Counting words in the Odyssey

“Tell me, O muse, of that ingenious hero who travelled far and wide after he had sacked the famous town of Troy…”

You know how to solve this! Code:

```python
counts = {}
with open('odyssey.txt', 'r') as f:
    for line in f.readlines():
        words = line.split()
        for word in words:
            counts[word] += 1
```

What if we gave you 25 computers instead of 1? Could you do it faster?
Counting words using MapReduce

**Map** - for every word, output (word, 1) to a file

```
[('tell', 1), ('me', 1), ('O', 1), ..., ('Troy', 1)]
```

**Group by key** - group every pair by the corresponding word

```
[('tell', [1, 1, 1]), ..., ('Troy', [1, 1, 1, 1, 1, 1, 1])]
```

**Reduce** - sum the list of 1's for each key to get count

```
[('tell', 3), ..., ('Troy', 7)]
```
Why is this so powerful?

Steps 1 and 3 are done in parallel! It runs way faster.

You don’t need to think about the implementation details.

Google used MapReduce to index the web and power its search algorithms, but distributed computing is everywhere.
Cloud Computing

What is it?

- On-demand access to shared computing resources
- Infrastructure/platform/software as a service
- Hardware virtualization

You use it in this class for all your programming assignments!

- Your Jupyter Notebooks aren’t running on your own computer - they’re on an Amazon or Google server somewhere!

Amazon Web Services, Google Cloud, Microsoft Azure

Companies spend hundreds of millions of dollars a year on it
Readings

New Yorker article about MapReduce:
https://www.newyorker.com/magazine/2018/12/10/the-friendship-that-made-google-huge

Articles about the Netflix Prize: https://www.thrillist.com/entertainment/nation/the-netflix-prize


Summary of data breaches:

Apps tracking your location: