Big Data Analytics: What is Big Data?

H. Andrew Schwartz
Stony Brook University
CSE545, Fall 2017
What’s the BIG deal?!
What’s the BIG deal?!
What’s the BIG deal?!

Peak of Inflated Expectations

Flu Trends Criticized (2014)

Plateau of Productivity

Slope of Enlightenment

Trough of Disillusionment


TIME

(Gartner Hype Cycle)
What's the BIG deal?!

Where are we today?

- Main-stream study being established
  - Realization of what subfields are really doing “big data” (i.e. data mining, ML, Statistics, computational social sciences).
  - Best practices being synthesized.

(Gartner Hype Cycle)
What’s the BIG deal?!

**Figure 3: Main challenges with big data projects**

What are the main challenges to implementing big data in your company?

- Security: 51%
- Budget: 47%
- Lack of talent to implement big data: 41%
- Lack of talent to run big data and analytics on an ongoing basis: 37%
- Integration with existing systems: 35%
- Procurement limitations on big data vendors: 33%
- Enterprise not ready for big data: 27%

*Source: Accenture Big Success with Big Data Survey, April 2014*
What’s the BIG deal?!

**Figure 6: Big data’s competitive significance**

- **Big data will revolutionize the way we do business to a degree similar to the advent of the Internet in the 1990s**
  - Strongly Agree: 51%
  - Agree: 38%
  - Neither Agree nor Disagree: 10%
  - Disagree: 1%

- **Big data will dramatically change the way we do business in the future**
  - Strongly Agree: 39%
  - Agree: 46%
  - Neither Agree nor Disagree: 13%
  - Disagree: 2%

- **Companies that do not embrace big data will lose their competitive position and may even face extinction**
  - Strongly Agree: 37%
  - Agree: 42%
  - Neither Agree nor Disagree: 19%
  - Disagree: 2%

- **We feel we are ahead of our peers in using big data and this creates a competitive advantage for us**
  - Strongly Agree: 37%
  - Agree: 46%
  - Neither Agree nor Disagree: 12%
  - Disagree: 4%

Source: Accenture Big Success with Big Data Survey, April 2014
What is Big Data?
What is Big Data?

data that will not fit in main memory.

traditional
computer science
What is Big Data?

data that will not fit in main memory.
data with a *large* number of observations and/or features.
What is Big Data?

data that will not fit in main memory.

data with a *large* number of observations and/or features.

non-traditional sample size (i.e. > 100 subjects); can’t analyze in stats tools (Excel).
What is Big Data? Industry view:

**Figure 2: Sources of big data**

Which of the following do you consider part of big data (regardless of whether your company uses each)?

- Large data files (20 terabytes or larger): 65%
- Advanced analytics or analysis: 60%
- Data from visualization tools: 50%
- Data from social networks: 48%
- Unstructured data (e.g., video, open text, voice): 43%
- Geospatial/location information: 38%
- Social media/monitoring/mapping: 37%
- Telematics: 34%
- Unstructured data/log files/free text: 25%

Source: Accenture Big Success with Big Data Survey, April 2014
What is Big Data? Industry view:

- Large data files (20 terabytes or larger): 65%
- Advanced analytics or analysis: 60%
- Data from visualization tools: 50%
- Data from social networks: 48%
- Unstructured data (e.g., video, open text, voice): 43%
- Geospatial/location information: 38%
- Social media/monitoring/mapping: 37%
- Telematics: 34%
- Unstructured data/log files/free text: 25%

Source: Accenture Big Success with Big Data Survey, April 2014
What is Big Data? Government view:

1. Survey of SDG-related Big Data projects

Type of data source(s)

- Mobile phone data: 23
- Satellite imagery data and geodata: 20
- Web data: 17
- Twitter data: 12
- Other social networks: 12
- Financial transaction data: 11
- Scanner data: 11
- Facebook data: 8
- Sensor data: 6
- Smart meter data: 5
- Health records: 2
- Ships identification data: 2
- Public transport usage data: 2
- Credit card data: 2

- Mobile (23), Satellite imagery (20) and social media (12+12+8) are the most prominent sources
What is Big Data?

**Short Answer:**

Big Data ≈ Data Mining ≈ Predictive Analytics ≈ Data Science (Leskovec et al., 2014)

**This Class:**

How to analyze data that is mostly too large for main memory.

Analyses only possible with a large number of observations or features.
What is Big Data?

**Goal:** Generalizations
A *model* or *summarization* of the data.

- How to analyze data that is mostly too large for main memory.
- Analyses only possible with a *large* number of observations or features.
What is Big Data?

Goal: Generalizations
A model or summarization of the data.

E.g.
- Google’s PageRank: *summarizes* web pages by a single number.
- Twitter financial market predictions: *Models* the stock market according to shifts in sentiment in Twitter.
- Distinguish tissue type in medical images: *Summarizes* millions of pixels into clusters.
- Mental Health diagnosis in social media: *Models* presence of diagnosis as a distribution (a summary) of linguistic patterns.
- Frequent co-occurring purchases: *Summarize* billions of purchases as items that frequently are bought together.
What is Big Data?

**Goal:** Generalizations
A *model* or *summarization* of the data.

1. **Descriptive analytics**
   Describe (generalizes) the data itself

2. **Predictive analytics**
   Create something *generalizeable* to new data
Big Data Analytics -- The Class

Core Data Science Courses
- CSE 519: Data Science Fundamentals
- CSE 544: Prob/Stat for Data Scientists
- **CSE 545: Big Data Analytics**
- CSE 512: Machine Learning
- CSE 537: Artificial Intelligence
- CSE 548: Analysis of Algorithms
- CSE 564: Visualization

Applications of Data Science
- CSE 507: Computational Linguistics
- CSE 527: Computer Vision
- CSE 549: Computational Biology
Big Data Analytics -- The Class

**Core Data Science Courses**
- CSE 519: Data Science Fundamentals
- CSE 544: Prob/Stat for Data Scientists
- **CSE 545: Big Data Analytics**
- CSE 512: Machine Learning
- CSE 537: Artificial Intelligence
- CSE 548: Analysis of Algorithms
- CSE 564: Visualization

**Applications of Data Science**
- CSE 507: Computational Linguistics
- CSE 527: Computer Vision
- CSE 549: Computational Biology

**Key Distinction:**
Focus on scalability and algorithms / analyses not possible without large data.
Big Data Analytics -- The Class

We will learn:

- to analyze different types of data:
  - high dimensional
  - graphs
  - infinite/never-ending
  - labeled
- to use different models of computation:
  - MapReduce
  - streams and online algorithms
  - single machine in-memory
  - Spark

Big Data Analytics -- The Class

We will learn:

● **to solve real-world problems**
  ○ Recommendation systems
  ○ Market-basket analysis
  ○ Spam and duplicate document detection
  ○ *Geo-coding data*

● **uses of various “tools”**:  
  ○ linear algebra
  ○ optimization
  ○ dynamic programming
  ○ hashing
  ○ *functional programming*
  ○ *tensorflow*

Big Data Analytics -- The Class

http://www3.cs.stonybrook.edu/~has/CSE545/
Preliminaries

Ideas and methods that will repeatedly appear:

- Bonferroni's Principle
- Normalization (TF.IDF)
- Hash functions
- IO Bounded (Secondary Storage)
- Power Laws
- Unstructured Data
Statistical Limits

Bonferroni's Principle
Statistical Limits

Bonferroni's Principle
Statistical Limits

Bonferroni's Principle

Which iphone case will be least popular?
Statistical Limits

Bonferroni's Principle

Which iPhone case will be least popular?

First 10 sales come in:

Can you make any conclusions?
Statistical Limits

Bonferroni's Principle

- Red
- Green
- Blue
- Teal
- Purple
- Yellow
Statistical Limits

Bonferroni's Principle
Statistical Limits

Bonferroni's Principle

Roughly, calculating the probability of any of n *findings* being true requires n times the probability as testing for 1 finding.

https://xkcd.com/882/

In brief, one can only look for so many patterns (i.e. features) in the data before you find something just by chance. “Data mining” was originally a bad word!
Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: TF.IDF
Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: TF.IDF of word $i$ in document $j$:

**Term Frequency:**

$$tf_{ij} = \frac{count_{ij}}{\max_k count_{kj}}$$

**Inverse Document Frequency:**

$$idf_i = \log_2 \left( \frac{\text{docs}_{*}}{\text{docs}_i} \right) \propto \frac{1}{\frac{\text{docs}_i}{\text{docs}_{*}}}$$

$$tf.idf_{ij} = tf_{ij} \times idf_i$$

where docs is the number of documents containing word $i$. 
Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: **TF.IDF** of word $i$ in document $j$:

**Term Frequency:**

$$tf_{ij} = \frac{count_{ij}}{\max_k count_{kj}}$$

**Inverse Document Frequency:**

$$idf_i = \log_2\left(\frac{docs_*}{docs_i}\right) \propto \frac{1}{\frac{docs_i}{docs_*}}$$

where $docs$ is the number of documents containing word $i$. 
Normalizing

**Standardize:** puts different sets of data (typically vectors or random variables) on the same scale with the same center.

- Subtract the mean (i.e. “mean center”)
- Divide by standard deviation

\[ z_i = \frac{x_i - \bar{x}}{S_x} \]
Hash Functions and Indexes

Review:

\( h: \text{hash-key} \rightarrow \text{bucket-number} \)

Objective: send the same number of expected hash-keys to each bucket

Example: storing word counts.
Hash Functions and Indexes

Review:

\[ h: \text{hash-key} \rightarrow \text{bucket-number} \]

Objective: send the same number of expected hash-keys to each bucket

Example: storing word counts.

\[ h(word) = \left( \sum_{\text{char} \in word} \text{ascii(char)} \right) \mod \#\text{buckets} \]
Hash Functions and Indexes

Review:

$h$: hash-key $\rightarrow$ bucket-number

Objective: send the same number of expected hash-keys to each bucket

Example: storing word counts.

\[
h(word) = \left( \sum_{\text{char} \in \text{word}} \text{ascii(char)} \right) \mod \#\text{buckets}
\]

Data structures utilizing hash-tables (i.e. O(1) lookup; dictionaries, sets in python) are a friend of big data algorithms! Review further if needed.
Hash Functions and Indexes

Review:

\[ h: \text{hash-key} \rightarrow \text{bucket-number} \]

Objective: send the same number of expected hash-keys to each bucket

Example: storing word counts.

Database Indexes: Retrieve all records with a given value. (also review if unfamiliar / forgot)

Data structures utilizing hash-tables (i.e. O(1) lookup; dictionaries, sets in python) are a friend of big data algorithms! Review further if needed.
IO Bounded

Reading a word from disk versus main memory: $10^5$ slower!

Reading many contiguously stored words is faster per word, but fast modern disks still only reach 150MB/s for sequential reads.

IO Bound: biggest performance bottleneck is reading / writing to disk.

(starts around 100 GBs; ~10 minutes just to read).
Power Law

Characterized many frequency patterns when ordered from most to least:

- County Populations [r-bloggers.com]
- # links into webpages [Broader et al., 2000]
- Sales of products [see book]
- Frequency of words [Wikipedia, “Zipf’s Law”]

(“popularity” based statistics, especially without limits)
Power Law

Power Law: \[ \log y = b + a \log x \]

raising to the natural log:

\[ y = e^b e^{a \log x} = e^b x^a = cx^a \]

where \( c \) is just a constant

Characterizes “the Matthew Effect” -- the rich get richer
Power Law

message-level

user-level

county-level
Data

Structured

Unstructured

- Unstructured ≈ requires processing to get what is of interest
- Feature extraction used to turn unstructured into structured
- Near infinite amounts of potential features in unstructured data
Data

Structured

mysql table  email header  satellite imagery  images
vectors matrices  facebook likes

Unstructured

● Unstructured ≈ requires processing to get what is of interest
● Feature extraction used to turn unstructured into structured
● Near infinite amounts of potential features in unstructured data