“Big data will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus.”

– McKinsey

Data Scientist: “The Sexiest Job of the 21st Century”

Why is data science booming as a field?
Everybody is selling big data
What is Big Data?

high **volume, velocity and/or variety** information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation. (Gartner)

1 Terabyte = 1024 Gigabytes
1 Petabyte = 1024 Terabytes
1 Exabyte = 1024 Petabytes
1 Zettabyte = 1024 Petabytes
Who does big data?

IBM, HP, EMC, Teradata, Oracle, SAP, Microsoft, AWS, Vmware, Google, Splunk, MemSQL, Palantir, Trifacta, Datameer, Neo, DataStax, Infobright, Fractal Analytics

http://www.datamation.com/applications/30-big-data-companies-leading-the-way-1.html
Big Data

- **Big $n$ vs. big $p$**
- **How is big data different?**
  - use available large-scale data rather than hoping for annotated data
- **Semi-parametric or non-parametric methods**
Different methods work best at scale

- Confusion set disambiguation
  - Choose the correct word in the set given the context

{principle, principal}
{then, than}
{to, two, too}
{weather, whether}

Figure 1. Learning Curves for Confusion Set Disambiguation
The unreasonable effectiveness of data

- Scene completion using millions of photographs
How to handle big data?

- Dimensionality reduction
- Sampling
- Hadoop/MapReduce
Big Data: different philosophy

Different data handling:
- Mostly unstructured data objects (Schema-less NoSQL)
- Vast number of attributes and data sources
- Data sources added and/or updated frequently
- Quality is unknown

Different programming philosophy:
- Distributed Programming
- Fault Tolerant

External References
http://developer.yahoo.com/hadoop/
http://code.google.com/edu/parallel/mapreduce-tutorial.html
What is the slowest part of big data analysis?

A) Multiplying X’X
B) Inverting a matrix (X’X)^-1?
C) Reading from X from disk to memory?
D) Other?
Data Parallelism

$P'' = P' + \Delta P$

Parameter Servers

Model Replicas

Data
Model Parallelism

Synchronous Variant
Map-Reduce Dataflow

Data is divided across multiple machines ("mappers")

Each mapper does the same thing to different data

Results are combined ("reduced")

How easy is it to do in map-reduce?

- Linear regression
- Linear regression with feature selection
- SVM
- k-NN
- K-means / EM

A) Easy
B) Hard
C) Impossible
Good tools

- LDA
  - Mallet
  - Factorie
- Deep Nets
  - Theano
  - Caffe, Torch
  - Tensorflow
Hadoop

MapReduce: Hadoop’s Original Data Processing Engine

Key Advances by MapReduce:

- **Data Locality:** Automatic split computation and launch of mappers appropriately
- **Fault-Tolerance:** Write out of intermediate results and restartable mappers meant ability to run on commodity hardware
- **Linear Scalability:** Combination of locality + programming model that forces developers to write generally scalable solutions to problems
In Hadoop

- **Hive**
  - data warehouse infrastructure providing data summarization, query, and analysis.

- **Pig, Crunch**
  - high-level platform for creating MapReduce programs used with Hadoop, pipelining code

- **Mahout**
  - scalable machine learning and data mining

- **Solr**
  - enterprise search platform built on Apache Lucene

- **Hue**
  - visualization
Spark

◆ Combines SQL, streaming, and complex analytics
◆ Often runs on Hadoop
  ● or Mesos, or standalone, or in the cloud
◆ Bindings to
  ● Java, Scala, Python, R
  ● NLTK …
◆ MLlib Machine Learning Library
  ● Faster than Mahout

Seems to be replacing Hadoop
Increasingly use a “deep stack”
Increasing in the cloud

◆ X as a Service
  - SaaS (software)
  - PaaS (platform)
  - IaaS (infrastructure)

◆ It’s easy to spin these up on AWS
  - Or MS Azure …
Tools are changing rapidly

- Currently hot:
  - SMACK: Spark, Mesos, Akka, Cassandra and Kafka
    - Spark – fast engine for distributed large-scale data processing
    - Mesos - distributed systems kernel
    - Akka - toolkit and runtime for building highly concurrent, distributed, and resilient message-driven applications
    - Cassandra – distributed database
    - Kafka - distributed publish-subscribe messaging system
  - Tensorflow

But the fundamentals we learned in this class are not changing!
Shiry Ginosaur et al.