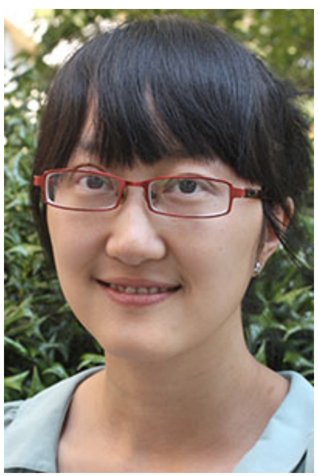
Algorithms for Big-Data Management

CompSci 590.04
Instructor: Ashwin Machanavajjhala

Duke UNIVERSITY

Course Staff

Ashwin Machanavajjhala
(Instructor)
Office Hours: By Appointment



Xi He
(Teaching Assistant)
Office Hours: Wed 4:30 – 6 PM

Duke UNIVERSITY

Administrivia

http://www.cs.duke.edu/courses/fall15/compsci590.4/

- Wed/Fri 3:05 – 4:20 PM
- "Reading Course + Project"
  - No exams!
  - Every class based on 1 (or 2) assigned papers that students *must* read.
- Projects: (50% of grade)
  - Individual or groups of size 3-4
- Assignments: (30% of grade)
  - There will be 3 assignments
- Class Participation: (other 20%)

Duke UNIVERSITY

Administrivia

- Projects: (50% of grade)
  - Ideas will be posted in the coming weeks
- Goals:
  - Literature review
  - Some original research/implementation
- Timeline (details will be posted on the website soon)
  - Sep 25: Choose Project (ideas will be posted … new ideas welcome)
  - Oct 2: Project proposal (1-4 pages describing the project)
  - Oct 30: Mid-project review (2-3 page report on progress)
  - Nov 20: Final presentations and submission (6-10 page conference style paper + 15minute talk)

Why you should take this course?

- Industry, academic and government research identifies the value of analyzing large data collections in all walks of life.
  - *"What Next? A Half-Dozen Data Management Research Goals for Big Data and Cloud", Surajit Chaudhuri, Microsoft Research*
  - *"Big data: The next frontier for innovation, competition, and productivity", McKinsey Global Institute Report, 2011*

Why you should take this course?

- Very active field and tons of interesting research.
  We will read papers in:
  - *Databases*
  - *Distributed Systems*
  - *Theory*
  - *Machine Learning*
  - *Privacy/Security*
  - *…*

Why you should take this course?

- Intro to research by working on a cool project
  - *Read scientific papers*
  - *Formulate a problem*
  - *Perform a scientific evaluation*

Today

- Course overview

- An algorithm for sampling

INTRODUCTION

What is Big Data?

http://visual.ly/what-big-data

http://visual.ly/what-big-data

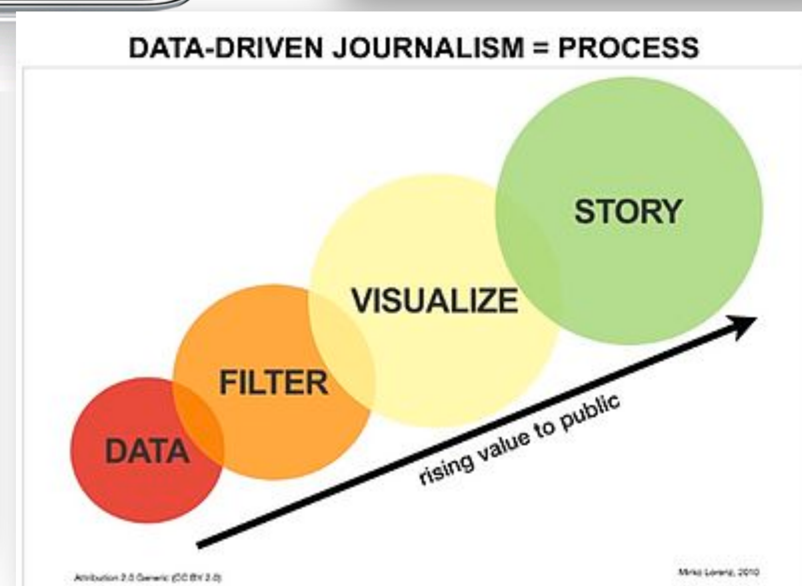
Duke UNIVERSITY

3 Key Trends

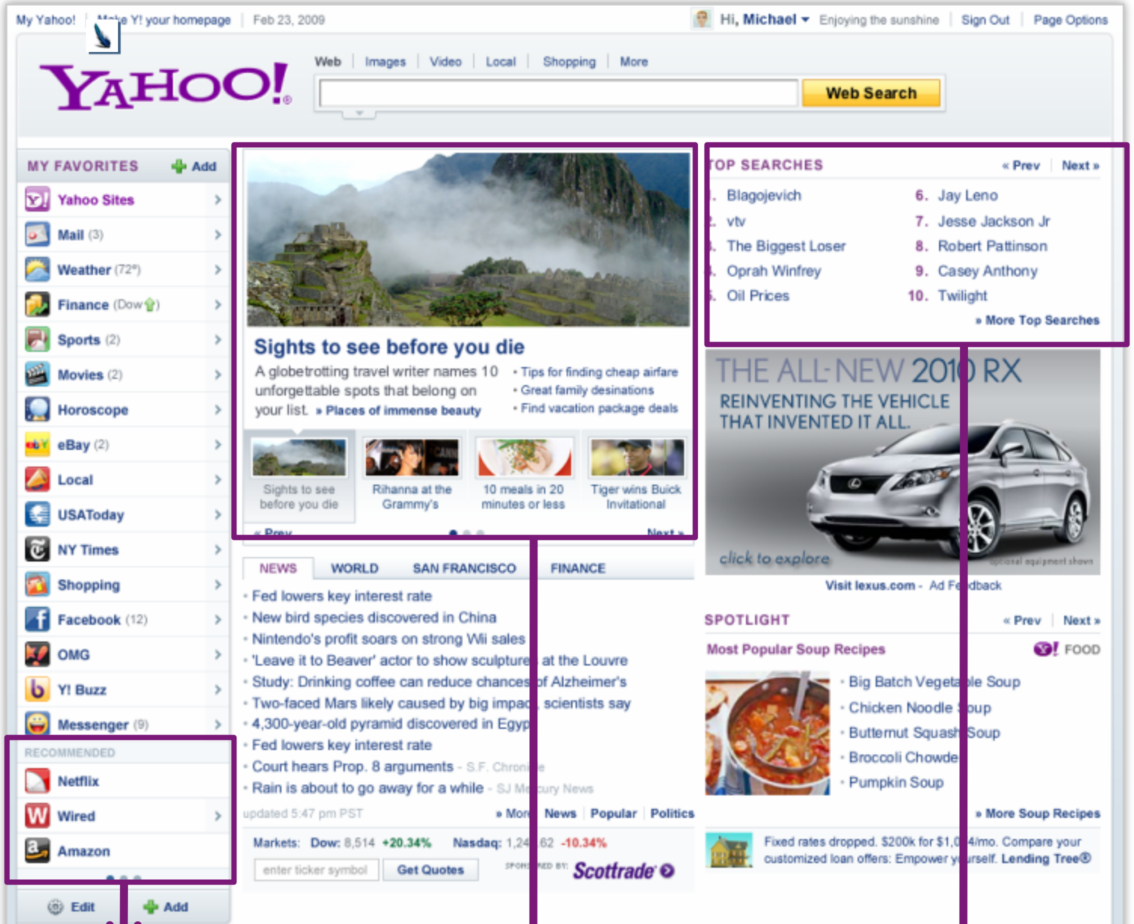
- Increased data collection
- (Shared nothing) Parallel processing frameworks on commodity hardware
- Powerful analysis techniques at both the population and individual levels

Duke UNIVERSITY

Big-Data impacts all aspects of our life



The value in Big-Data …



Recommended links

+79% clicks

vs. randomly selected

Personalized News Interests

+250% clicks

vs. editorial one size fits all

Top Searches

+43% clicks

vs. editor selected

The value in Big-Data …

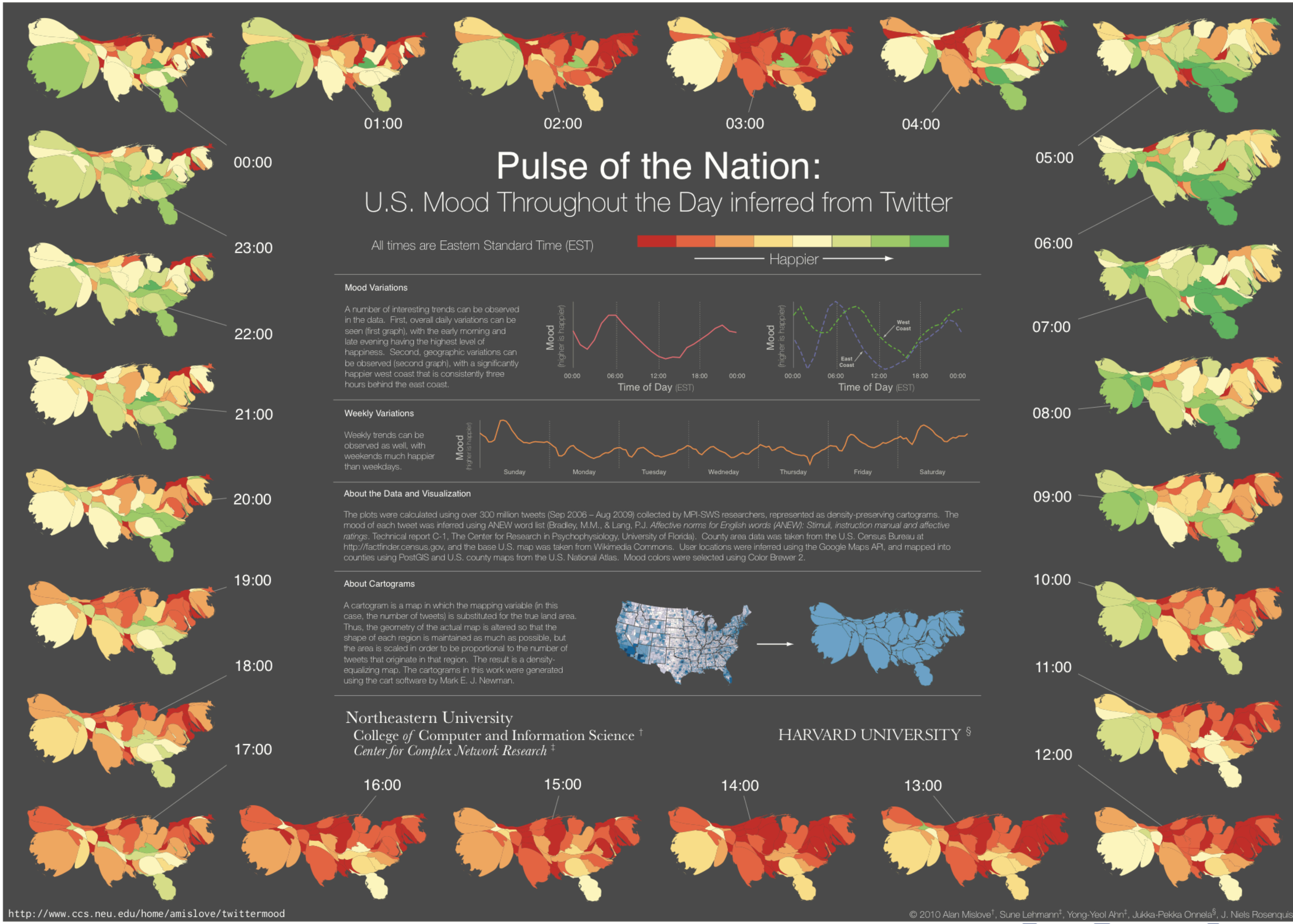
"*If* ***US healthcare*** *were to use* ***big data*** *creatively and effectively to drive efficiency and quality, the sector could create more than* ***$300 billion in value every year****.*"

McKinsey Global Institute Report

Duke UNIVERSITY

Example: Google Flu

Duke UNIVERSITY



http://www.ccs.neu.edu/home/amislove/twittermood/

Course Overview

We will learn strategies for handling data that is …

1. large
2. fast
3. sensitive
4. partitioned

… and along the way we will learn a number of useful tricks.

Course Overview

Strategy 1: Compute approximate answers on *large data*

- Sampling
  - Reservoir Sampling
  - Sampling with indices/Joins
  - Monte Carlo method
  - Markov Chains

Course Overview

Strategy 2: Compute approximate answers *on fast data*

- Streaming
  - Sketches
  - Online Aggregation
  - Online learning

Duke UNIVERSITY

Course Overview

Strategy 3: Throw a lot of hardware at *large data*

- Parallel Architectures & Algorithms
  - Map Reduce
  - Graph processing architectures : Bulk Synchronous parallel and asynchronous models
  - (Graph connectivity, Matrix Multiplication, Belief Propagation

Course Overview

Strategy 4: Add noise to handle *sensitive data*

- Computing under noise
  - Differential privacy
  - Histograms
  - Range queries
  - Sorting

Course Overview

Strategy 5: Join *partitioned data*

- Joining datasets & Record Linkage
  - Theta Joins: or how to optimally join two large datasets
  - Clustering similar documents using minHash
  - Correlation Clustering

SAMPLING

Why Sampling?

- Approximately compute quantities when
  - Processing the entire dataset takes too long.
    How many tweets mention Obama?
  - Computation is intractable
    Number of satisfying assignments for a DNF.
  - Do not have access or expensive to get access to entire data.
    How many restaurants does Google know about?
    Number of users in Facebook whose birthday is today.
    What fraction of the population has the flu?

Zero-One Estimator Theorem

Input: A universe of items U (e.g., all tweets)
A subset G (e.g., tweets mentioning Obama)

Goal: Estimate **$\mu = |G|/|U|$**

Algorithm:

- Pick N samples from U {x1, x2, …, xN}
- For each sample, let Yi = 1 if xi ε G.
- Output: Y = Σ Yi/N

Theorem: Let $\varepsilon < 1.5$. If $N > (1/\mu)\,(3 \ln(2/\delta)/\varepsilon^2)$, **then**

$$\mathbf{Pr[(1-\varepsilon)\,\mu < Y < (1+\varepsilon)\mu] > 1-\delta}$$

Zero-One Estimator Theorem

Algorithm:

- Pick N samples from U {x1, x2, …, xN}
- For each sample, let Yi = 1 if xi ε G.
- Output: Y = Σ Yi/N

Theorem: Let $\varepsilon < 1.5$. If $N > (1/\mu)\ (3 \ln(2/\delta)/\varepsilon^2)$, **then**

$$\Pr[(1-\varepsilon)\,\mu < Y < (1+\varepsilon)\mu] > 1-\delta$$

Proof: Homework

Estimating multiple properties

- Suppose there are 'm' subsets of interest G1, G2, …, Gm
- Goal: Estimate **μi = |Gi|/|U|** for all i

- How many samples do we need?

Estimating multiple properties

- Suppose there are 'm' subsets of interest G1, G2, …, Gm
- Goal: Estimate **$\mu i = |Gi|/|U|$** for all i

- How many samples do we need?

- Answer: $N > (3/\mu\varepsilon^2)(\ln m + \ln(2/\delta))$, where $\mu = \min_i \mu i$

Simple Random Sample

- Given a table of size N, pick a subset of n rows, such that each subset of n rows is equally likely.

- How to sample n rows?
- … if we don't know N?

Reservoir Sampling

Highlights:

- Make one pass over the data
- Maintain a reservoir of n records.
- After reading t rows, the reservoir is a simple random sample of the first t rows.

Reservoir Sampling [Vitter ACM ToMS '85]

Algorithm R:

- Initialize reservoir to the first n rows.

- For the $(t+1)^{st}$ row R,

  – Pick a random number m between 1 and t+1

  – If m <= n, then replace the m^{th} row in the reservoir with R

Proof

Proof

- If N = n, then P [row is in sample] = 1. Hence, reservoir contains all the rows in the table.

- Suppose for N = t, the reservoir is a simple random sample. That is, each row has n/t chance of appearing in the sample.

- For N = t+1:
  - (t+1)st row is included in the sample with probability n/(t+1)
  - Any other row:
    P[row is in reservoir] = P[row is in reservoir after t steps]* P[row is not replaced]
    = n/t * (1-1/(t+1)) = n/(t+1)

Duke UNIVERSITY

Complexity

- Running time: O(N)
- Number of calls to random number generator: O(N)

- Expected number of elements that may appear in the reservoir:

$$n + \Sigma_n^{N-1} n/(t+1) = n(1 + H_N - H_n) \approx n(1 + \ln(N/n))$$

- Is there a way to sample faster? in time O(n(1 + ln(N/n))) ??

Faster algorithm

- Algorithm R skips over (does not insert into reservoir) a number of records ($N - n(1 + \ln(N/n))$)

- At any step t, let S(n,t) denote the number of rows skipped by the Algorithm R.
  - Involved O(S) time and O(S) calls to the random number generator.

- $P[\, S(n,t) = s \,] = ?$

Duke UNIVERSITY

Faster algorithm

- At any step t, let $S(n,t)$ denote the number of rows skipped by the Algorithm R.

- $P[\ S(n,t) = s\ ]$ = for all $t < x <= t+s$, row x was not inserted into reservoir, but row t+s+1 is inserted.
  $= \{ 1-n/(t+1) \} \times \{1 - n/(t+2)\} \times \ldots \times \{1-n/(t+s)\} \times n/(t+s+1)$

- We can derive expression for CDF:
  $P[\ S(n,t) <= s\ ] = 1 - (t/t+s+1)(t-1/t+s)(t-2/t+s-1) \ldots (t-n+1/t+s-n+2)$

Faster Algorithm

Algorithm X

- Initialize reservoir with first n rows.
- After seeing t rows, randomly sample a skip s = S(n,t) from the CDF
- Pick a number m between 1 and n
- Replace the mth row in the reservoir with the (t+s+1)st row.
- Set t = t + s + 1

Faster Algorithm

Algorithm X

- Initialize reservoir with first n rows.
- After seeing t rows, randomly sample a skip s = S(n,t) from the CDF
  – Pick a random U between 0 and 1
  – Find the minimum s such that P[S(n,t) <= s] <= 1-U
- Pick a number m between 1 and n
- Replace the mth row in the reservoir with the (t+s+1)st row.
- Set t = t + s + 1

Algorithm X

- Running time:
  Each skip takes O(s) time to compute
  Total time = sum of all the skips = O(N)

- Expected number of calls to the random number generator
  = 2 * expected number of rows in the reservoir
  = $O(n(1 + \ln(N/n)))$ optimal!

See paper for algorithm which has optimal runtime

Duke UNIVERSITY

Summary

- Sampling is an important technique for computation when data is too large, or the computation is intractable, or if access to data is limited.

- Reservoir sampling techniques allow computing a sample even without knowledge of the size of the data.
  - Also can do weighted sampling [Efraimidis, Spirakis IPL 2006]

- Very useful for sampling from streams (e.g., twitter stream)

Duke UNIVERSITY

References

- J. Vitter, “Random Sampling with a Reservoir”, ACM Transaction on Mathematical Software, 1985
- P. Efraimidis, P. Spirakis, “Weighted random sampling with a reservoir”, Journal Information Processing Letters, 97(5), 2006
- R. Karp, R. Luby, N. Madras, “Monte Carlo Approximation Algorithms for Enumeration Problems”, Journal of Algorithms, 1989