Algorithms for Data Science

Barna Saha, David Wemhoener

Fall 2017

A new algorithms class!

- Why do we need a new algorithms class?
  - Unprecedented amount of data containing a wealth of information.
    - Example: Twitter receives 6000 tweets per second which amounts to 500 million tweets per day with a storage requirement of ~7000 gigabytes.
  - Traditional algorithms process data in RAM, sequentially and may have high time-complexity
    - Not suitable for processing Twitter data

Characteristics of Big Data

- VOLUME
  - Can not store the entire data in the main memory
- VELOCITY
  - Data changes frequently. Needs highly efficient processing, often parallel processing.
- VARIETY & VERACITY
  - Data coming from many different sources, and often contains noise-adds to the complexity of data processing

This Course

- Develop algorithms to deal with such data
  - Space and Time Efficient
  - Parallel Processing
  - Approximation & Randomization
- Theoretical course with main focus on algorithm analysis
  - Relevant applications will be discussed, and there will be plenty of coding exercises
  - But no software tools will be covered
- Background in basic algorithms (311) and probability (240) are strictly required.

Personnel

- Instructors
  - Barna Saha
    - Email: barna@cs.umass.edu
    - Office Hour: Tue 11:30-12:30, CS336
  - David Wemhoener
    - Email: wem@cs.umass.edu
    - Office Hour: Thur 2:00-3:00 pm, CS XXX
- Teaching Assistants
  - Sainyam Galhotra
    - Email: sainyam@cs.umass.edu
    - Office Hour: Wed 2:00-3:00 pm, CS207
  - Raghavendra Addanki
    - Email: raddanki@cs.umass.edu
    - Office Hour: Mon 4:30-5:30 pm, CS207

Grading

- Homeworks (4) in a group of 2/3
  - Will consist of mathematical problems and/or programming assignments
  - Find your partners early and wisely. Do not come to me with complaints about your partner.
  - 30%
- Midterm
  - 20%
- Final
  - 30%
- Mini Coding/Programming Assignments
  - Few simple exercises to be done in a group of 2/3
  - Roughly 4
  - 20%

Communication

- All class related discussions should be done through piazza.
  - Sign up from the course page.
- Course website
  - http://www-edlab.cs.umass.edu/cs590d/
- Homework submission
  - Must be submitted in hardcopy at the CS main office dropbox
  - All codes must be submitted via Moodle
  - Absolutely no submission by email

Books

- Text Book: We will use reference materials from the following books. **Both can be downloaded for free.**
- Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman and Jeff Ullman.
- Foundations of Data Science, a book in preparation, by John Hopcroft and Ravi Kannan

An Interesting Problem

- Suppose we see a sequence of items, one at a time.
- We want to keep a single item in memory.
- We want it to be selected at random from the sequence.
- Easy if we know the number of items "n"
  - Just draw a random number in between 1 and n
- What if we do not know n?

Reservoir Sampling

- Upon seeing the **first** element—keep it.
  - The first element is chosen with probability 1.
- Upon seeing the **second** element—select the second element with probability $\frac{1}{2}$. If the second element is selected discard the first element.
  - The probability that the second item is sampled=$\frac{1}{2}$
  - The probability that the first item is sampled=$\frac{1}{2}$
- Upon seeing the **third** element—select it with probability $\frac{1}{3}$, if selected then discard the element that was previously selected.
  - The probability that the third item is sampled=$\frac{1}{3}$.
  - The probability that the second item is sampled=$\frac{2}{3} * \frac{1}{2} = \frac{1}{3}$
  - The probability that the first item is sampled=$\frac{2}{3} * \frac{1}{2} = \frac{1}{3}$

Reservoir Sampling

- Upon seeing the **fourth** element—select it with probability $\frac{1}{4}$, if selected then discard the element that was previously selected.
  - The probability that the fourth item is sampled$=\frac{1}{4}$.
  - The probability that the third item is sampled$=\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$
  - The probability that the third item is sampled$=\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$ The probability that the third item is sampled$=\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$

- Can you generalize the algorithm to any i?

  - Upon seeing the ith item—select it with probability $\frac{1}{i}$, if selected, discard the element that was previously selected.
    - The probability that the ith item is sampled$=\frac{1}{i}$.
    - The probability that the jth $j < i$ item is sampled$=\frac{i-1}{i} * \frac{1}{i-1} = \frac{1}{i}$

Reservoir Sampling

- Upon seeing the **fourth** element—select it with probability $\frac{1}{4}$, if selected then discard the element that was previously selected.
  - The probability that the fourth item is sampled=$\frac{1}{4}$.
  - The probability that the third item is sampled=$\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$
  - The probability that the third item is sampled=$\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$ The probability that the third item is sampled=$\frac{3}{4} * \frac{1}{3} = \frac{1}{4}$

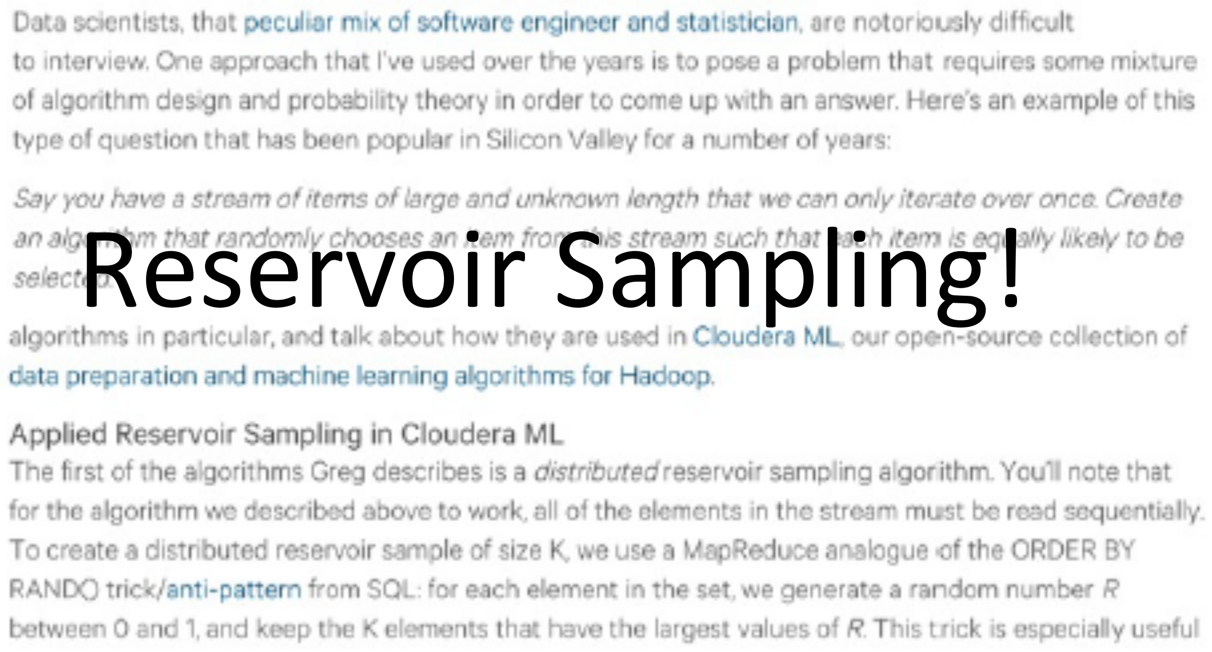
- Can you generalize the algorithm to any i?

  - Upon seeing the ith item—select it with probability $\frac{1}{i}$, if selected, discard the element that was previously selected.
    - The probability that the ith item is sampled=$\frac{1}{i}$.
    - The probability that the jth $j < i$ item is sampled=$\frac{i-1}{i} * \frac{1}{i-1} = \frac{1}{i}$

What happens when the reservoir can store "s" elements?

Algorithms Every Data Scientist Should Known: By Josh Wills

https://blog.cloudera.com/blog/2013/04/hadoop-stratified-randosampling-algorithm/



Data scientists, that peculiar mix of software engineer and statistician, are notoriously difficult to interview. One approach that I've used over the years is to pose a problem that requires some mixture of algorithm design and probability theory in order to come up with an answer. Here's an example of this type of question that has been popular in Silicon Valley for a number of years:

Say you have a stream of items of large and unknown length that we can only iterate over once. Create an alg[illegible]thm that randomly chooses an item fro[illegible] this stream such that [illegible]h item is eq[illegible]ally likely to be select[illegible]

algorithms in particular, and talk about how they are used in Cloudera ML, our open-source collection of data preparation and machine learning algorithms for Hadoop.

Applied Reservoir Sampling in Cloudera ML

The first of the algorithms Greg describes is a *distributed* reservoir sampling algorithm. You'll note that for the algorithm we described above to work, all of the elements in the stream must be read sequentially. To create a distributed reservoir sample of size K, we use a MapReduce analogue of the ORDER BY RAND() trick/anti-pattern from SQL: for each element in the set, we generate a random number *R* between 0 and 1, and keep the K elements that have the largest values of *R*. This trick is especially useful

Reservoir Sampling!

Sampling

- A very useful method to obtain appropriate summary of data
- Will learn more in the coming classes
- But needs to be done with care
- Link to video https://www.youtube.com/watch?v=xmhVdsOTh1E

Mini Exercise-1

- Implement reservoir sampling when reservoir has size 1. Let the items from 1 to 100 appear one by one.
  - Report the item sampled in one run of the algorithm.
  - Repeat the algorithm for 1000 times and plot the number of times each element is selected.
  - Repeat the algorithm for 10000 times and plot the number of times each element is selected.
  - Repeat the algorithm for 100000 times and plot the number of times each element is selected.

  - **DUE: Tuesday, 11th.**

Next Few Classes

- Probability review before we enter into the more interesting regime!