CSE 522: Sublinear (and Streaming) Algorithms **Spring 2014**

Lecture 1: Introduction to Sublinear Algorithms

March 31, 2014

Lecturer: Paul Beame *Scribe: Paul Beame*

Too much data, too little time, space for it all.

1 Sublinear Time

Why? It may be that not all data can be accessed, or the accessible data may simple be too large to spend time reading.

Since the algorithms can't access all the data, the answers can no longer be exact. Some problems are easy to approximate: average, or median.

What kinds of approximation?

- Produce an answer that is "close" to the true answer. For example, for an optimization problem produce a value that is close the optimum value OPT. E.g.

  $$OPT/c \leq \text{answer} \leq c \cdot OPT.$$

  Note: There is too little to write down a full solution.

- Produce an answer that is correct an input that is "close" to the given input.

  e.g. Property Testing for decision problems:
  Determine whether

  1. the input has the property, or
  2. the input is "far" from every input having the property.

  Don't care otherwise.

The algorithm may make random choices with the probability of producing an incorrect answer at most δ. In general, randomization will be essential to the algorithms we consider.

Sublinear Time Models

- **Random access queries** Can access any specific word of the input in one step.
- **Samples** Can get a word of input from a random position in one step (with that position), or a random sample from an input distribution in one step.

2 Sublinear Space/Streaming

Can view all the data but it is going by so fast that the algorithm cannot store it all.

This is particular useful for analyzing:
Network traffic
Database transactions
Sensor feeds
Scientific data

The relevant question is how to keep track of a summary of the data seen in order to answers, since there is no space to store it all.

In general, space complexity is at most the time complexity, so sublinear space is more general than sublinear time, but we only deal with restricted version of sublinear space.

Data Stream Model Single pass through the data: Stream $x_1, x_2, \ldots, x_n \in [M]$ arriving in order. Fast, per element processing. Amount of storage will be the main parameter. The goal will be to have space at most n^α, or even better, $\log^c n$.

There also is a variant of the data model that is useful in another context: The data either resides has been streamed onto disk first. The storage is the amount of main memory needed to process it using fast sequential access. Multiple passes in the same order may make sense for this variant.

Sketching for Data Streams Store a short *sketch*, or synposis, of the data for later use. Of particular use will be *linear sketching* which for input $x = (x_1, \ldots, x_n)$ stores the sketch Ax where A is an $m \times n$ matrix for $m \ll n$.

Linear sketching is also useful for *compressed sensing*, sublinear measurement where the sketch happens during input collection. We will probably not have time to cover this, however.

Course Details In general, this course will focus on common approaches and tools. The course website is http://www.cs.washington.edu/522. There is no text but the website http://sublinear.info which is linked on the course website has many resources, ranging from links to surveys and papers in the area, other courses on sublinear algorithms, to a list of open problems.

The course requirements will consist of problem sets (roughly 2) as well as a project presentation which consists of either presenting paper from a list that I will make available, or present results of experimental application of sublinear algorithm techniques.

3 Property Testing Example

We consider the *Element Distinctness* problem: Given $x_1, \ldots, x_n \in [M]$ for $M \geq n$, are all x_i distinct?

The property testing version of this problem is to distinguish between the cases:

1. All x_i are distinct
2. The number of distinct elements in x is $< (1-\varepsilon)n$

Either answer is OK if neither case holds.

To see how this fits into the usual property testing framework using the following definition:

Definition 3.1. $x, y \in [M]^n$ *are* ε-far *iff* $|\{i\ x_i \neq y_i\}| > \varepsilon n$.

Observe that the set of inputs that are ε-far from any distinct input is precise the set of inputs with $< (1-\varepsilon)n$ distinct elements.

Obvious algorithm Take s independent samples (remembering where they came from). If there is a duplicate in the sample, output FAIL else output PASS.

This algorithm always answers correctly on distinct inputs. If the input is not distinct, how many samples are needed to detect this with probability $\geq 1/2$?

Consider input examples with $\varepsilon \approx 1/2$:
If input is $1, 1, \ldots, 1, 2, \ldots, n/2$ in random order, only a constant number of samples suffice. However if input is $1, 1, 2, 2, 3, 3, \ldots, n/2, n/2$ in random order, what is the probability that a duplicate will be found?

For each $i \neq j$, the probability that sample j is the unique match for sample i is precisely $1/n$. Therefore

$$\mathbb{P}[\text{duplicate found}] \leq \mathbb{E}[\#\text{ duplicates found}] \leq \binom{s}{2}/n < \frac{s^2}{2n}$$

so we need $s > \sqrt{n}$ to get error $< 1/2$.

Probability Tools: Tail Bounds

Let $\mu = \mathbb{E}(x)$. Recall that $Var(X) = \mathbb{E}[(X-\mu)^2] = \mathbb{E}(X^2) - \mathbb{E}(X)^2$. We will use the following tail bounds repeatedly throughout the course.

Linear Tails: Markov's Inequality If X is a random variable and we always have $X \geq 0$, then for $k > 0$,

$$\mathbb{P}[X \geq k] \leq \mathbb{E}(X)/k.$$

Quadratic Tails: Chebyshev's Inequality For any random variable X,

$$\mathbb{P}[|X - \mathbb{E}(X)| \geq k] \leq Var(X)/k^2.$$

This follows easily by applying Markov's inequality to $(X-\mu)^2$ for $\mu = E(X)$. We will find it particularly useful in the case of pairwise independent random variables.

Exponential Tails: Bernstein/Chernoff/Hoeffding's Bounds These are used for the sums of independent bounded random variables.

In particular, we will use the so-called Chernoff Bounds for

$$X = X_1 + \cdots + X_n$$

where the X_i are independent $\{0,1\}$ random variables. (The X_i are not required to have the same distribution, so this is the more general case of Poisson trials rather than just the usual Bernoulli trials.)

$$\mathbb{P}[X \geq (1+\delta)\mathbb{E}(X)] \leq e^{-\delta^2\mathbb{E}(X)/3}, \qquad \text{and}$$
$$\mathbb{P}[X \leq (1-\delta)\mathbb{E}(X)] \leq e^{-\delta^2\mathbb{E}(X)/2}.$$

If we don't worry too much about the constants in the exponent, we can summarize this in a single inequality as

$$\mathbb{P}[|X - \mathbb{E}(X)| \geq \delta\mathbb{E}(X)] < e^{-\delta^2\mathbb{E}(X)/4}.$$

Analyzing the Obvious Algorithm

Since we know the algorithm is correct when the input is distinct, we consider the case when there are $< (1-\varepsilon)n$ distinct elements. We target correctness with probability at least 2/3.

MAIN IDEA: pair off the duplicates and argue that roughly $\sqrt{n}$ samples are likely to hit both members of some pair.

For convenience of analysis we consider $2s$ elements chosen in two phases with s samples chosen in each phase and let S_1 and S_2 denote the samples from the respective phases.

We will show that

- S_1 hits the 1st element from many pairs with probability at least 5/6
- Given that the first phase succeeds, S_2 hits the 2nd element from some pair whose 1st element is hit by S_1 with probability at least 5/6.

We can assume without loss of generality that $s \geq \sqrt{n}$ and for convenience we assume that $s \leq \varepsilon n/48$.

If the number of distinct elements is $< (1-\varepsilon)n$, then if we pair up duplicate values (ignoring one copy of any value that occurs an odd number of times) then we get $> \varepsilon n/2$ pairs of duplicates. (If we list one copy of each of the $< (1-\varepsilon)n$ distinct values first, if a value occurs k times among the remaining $> \varepsilon n$ elements, it will be part of $\lceil k/2 \rceil$ pairs, one involving the occurrence of that value among the first list of distinct values.) Therefore

$$\mathbb{P}[\text{a single sample hits 1st element of a pair}] > \varepsilon/2.$$

Let indicator variable $Y_i = \begin{cases} 1 & i^{\text{th}} \text{ sample in } S_1 \text{ hits 1st element of a pair} \\ 0 & \text{otherwise.} \end{cases}$

Then

$$\mathbb{E}[\#\text{ samples in } S_1 \text{ that hit 1st element of a pair}] = \sum_{i=1}^{s} Y_i > \varepsilon s/2.$$

The Y_i are independent so by the Chernoff bound

$$\mathbb{P}[\sum_{i=1}^{s} Y_i < \mathbb{E}(\sum_{i=1}^{s} Y_i)/2] \leq e^{-\mathbb{E}(\sum_{i=1}^{s} Y_i)/8} < e^{\varepsilon s/16} \ll 1/12.$$

Though we have $\mathbb{P}[\sum_{i=1}^{s} Y_i < \varepsilon s/2] \ll 1/12$, this is not quite enough to count the pairs hit by S_1 since the sample may hit the same pair more than once. We bound this by subtracting off the contribution to $\sum_{i=1}^{s} Y_i$ from such doubly hit pairs.

Let $Z_{ij} = \begin{cases} 1 & \text{sample } i \text{ and sample } j \text{ of } S_1 \text{ collide} \\ 0 & \text{otherwise.} \end{cases}$

As we saw earlier with the lower bound on s, we have $\mathbb{E}(Z_{ij}) = \mathbb{P}[Z_{ij}] = 1/n$ and hence the expected total number of colliding pairs from S_1 is at most

$$\mathbb{E}(\sum_{i \neq j} Z_{ij}) = \binom{s}{2}/n \leq \frac{s^2}{2n} < \frac{s}{2n}\frac{\varepsilon n}{48} = \varepsilon n/96.$$

Therefore by Markov's inequality we have

$$\mathbb{P}[\sum_{i \neq j} Z_{ij} > 12 \cdot \varepsilon s/96 = \varepsilon s/8] \leq 1/12.$$

Since we have failure at most 1/12 for the two parts, hence, except with probability at most 1/6, the total number of pairs of duplicates whose first element is hit by S_1 is $> \varepsilon s/4 - \varepsilon s/8 = \varepsilon s/8$.

Phase 2 The analysis of the second phase is now much easier assuming that the first phase succeeds; i.e., that there are $> \varepsilon s/8$ pairs of duplicates whose 1st element is hit by some sample in S_1.

Let H_1 be the set of pairs of duplicates whose 1st element is hit by some sample from S_1. Then

$$\mathbb{P}[j^{\text{th}} \text{ sample of } S_2 \text{ hits 2nd element of a pair in } H_1] > \frac{\varepsilon s}{8n}.$$

It follows that

$$\begin{aligned} \mathbb{P}[\text{no sample in } S_2 \text{ hits 2nd element of a pair in } H_1] &< \left(1 - \frac{\varepsilon s}{8n}\right)^s \\ &\leq \left(e^{-\frac{\varepsilon s}{8n}}\right)^s \\ &= e^{-\frac{\varepsilon s^2}{8n}} < 1/6, \end{aligned}$$

for $\frac{\varepsilon s^2}{8n} \geq 2$, where the second inequality follows using the fact that $1 - x \leq e^{-x}$ for all real values of x. Observing that $\frac{\varepsilon s^2}{8n} \geq 2$ is equivalent to $s \geq 4\sqrt{n/\varepsilon}$, we see that choosing $s = \lceil 4\sqrt{n/\varepsilon} \rceil$ suffices for the total probability that no duplicate is found to be at most 1/3.

Therefore $O(\sqrt{n/\varepsilon})$ samples suffice to test for Element Distinctness.

First Problem Set #1: Generalize the lower bound argument to show that any algorithm that always accepts distinct inputs, but with probability at least 1/2 rejects all inputs with $< (1 - \varepsilon)n$ distinct values requires $\Omega(\sqrt{n/\varepsilon})$ samples.