Randomized Algorithms 2020-1
Lecture 5

Streaming Algorithms *

Moni Naor

1 Multiset equality

Last time: We have two multi-sets A and B and they are given in an arbitrary order. Once an element is given it cannot be accessed again (unless it is explicitly stored) and our goal is to have a low memory algorithm. We required a family of functions H that was incremental in nature, in the sense that for a function $h \in H$:

- Given $h, h(A)$ and an element x it is easy to compute $h(A \cup \{x\})$.
- For any two different multi-sets A and B the probability over the choice of $h \in H$ that $h(A) = h(B)$ is small.
- The description of h is short and the output of h is small.

The function we saw was based on treating the set A as defining a polynomial $P_A(x) = \Pi_{a \in A}(x - a)$ over a finite field whose size is larger than the universe from which the elements of A are chosen (say a prime $Q > |U|$). The member of the family of functions is called h_x for $x \in GF[Q]$ and defined as $h_x(A) = P_A(x)$. The probability that two sets collide (i.e. $h_x(A) = h_x(B)$, which in turn means that $P_A(x) = P_B(x)$) is $\max\{|A|, |B|\}/Q$, since this is the maximum number of points that two polynomials whose degree is at most $\max\{|A|, |B|\}$ can agree without being identical.

Storing h_x and storing $h(A)$ as it is computed requires just $O(\log Q)$ bits, so the resulting algorithm never needs to store anything close size to the original sets.

2 Communication Complexity

Let $f : 0, 1^n \times \{0, 1\}^n \mapsto \{0, 1\}$. Communication complexity studies the length of the message that two parties A and B need to exchange to compute $f(x, y)$ where x is the input of A and y is the

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. In the interest of brevity, most references and credits were omitted.

input of B. The one-way Communication Complexity of f is that of a protocol where A sends a message to B and then B computes $f(x, y)$ on her own.

The point is that a one-pass, low-memory streaming algorithm on input (x, y) for computing $f(x, y)$ implies a one-way communication protocol for $f(x, y)$ and the communication is the memory size.

We can use this to show lower bounds on the amount of memory needed for various problems. The disjointness problem DISJ is very useful. Here we have a universe $\{1 \dots n\}$ and the inputs are two subsets $X, Y \subseteq \{1 \dots n\}$ and the question is whether they are disjoint or not.

Theorem 1. *Every randomized protocol that for every input (x, y) correctly decides the function DISJ with probability at least $2/3$, uses $\Omega(n)$ bits of communication in the worst case*

We used this to argue that the problem of deciding whether the set of elements labeled A and the set of elements labeled B are equal as sets (rather than multi-sets) requires $\Omega(n)$ bits of memory. The reduction from DISJ is based on two sets:

- $\{1, 2, ..., n\}$
- $\bar{X} \cup \bar{y}$ (the union of the complements of X and Y)

They are equal iff X and Y do not intersect.

There is a recent book on Communication Complexity. See Chapter 6 for the proof of the above theorem:

- Anup Rao and Amir Yehudayoff, **Communication Complexity and Applications**, Cambridge 2020.

See the lecture by O'Donnel on the topic:

- Lectures 23a-d of CS Theory Toolkit `https://www.youtube.com/watch?v=mQQ36cDnmR8`

3 Frequency Moments

We saw an elegant algorithm for computing F_2. The algorithm requires a four-wise independent hash function, that is a family of functions H s.t. for any four different elements x_1, x_2, x_3, x_4 the values $h(x_1), h(x_2), h(x_3), h(x_4)$ are independent, where the randomness is over the choice of $h \in H$.

Recall that we said that an estimator that estimates a value might not be very useful even if the expected value it returns equals the value we are after. The example was estimating the number of satisfying assignments a given formula has. This is a $\#P$-Complete problem. If there are n variables, choose uniformly at random an assignment in $\{0, 1\}^n$ and if the assignment satisfies the formula, then return 2^n and otherwise return 0. Argue that the expected value is correct. What is the variance? Why is it useless?

4 Homework

1. Consider the simultaneous message model for evaluating a function $f(x, y)$: Alice and Bob share a random string. They receive inputs x and y respectively and each should send a message to a referee, Charlie, who should evaluate the function $f(x, y)$. They may also have their own private source of randomness. The goal is for Alice and Bob to send short messages to Charlie.

   We will consider the equality function.

   (a) Consider first the case that Alice and Bob share a random string. the protocol where Alice and Bob send to Charlie an inner product of their input with a common random string r and Charlie decides based on the equality of the messages he receives. What happens to this protocol if Eve, who selects the inputs and whose goal is to make Charlie compute the wrong value, knows the common string r when she selects x and y?

   (b) Suggest a non-trivial protocol for this case, i.e. one whose communication complexity is sublinear in the input length.

2. Recall the memory checking problem, where a processor has access to a large memory and the goal is to discover whether the memory malfunctions. The memory malfunctioning means that it does not return the correct value that was last stored at a given location (i.e. the last value stored at that location). The processor is given a sequence of read operations (get the value stored at location a) and write operations (store value v at location a) and should either perform them as given or declare that the memory malfunctioned (but should do the latter only if the memory indeed malfunctioned). The memory is controlled by an adversary and the processor has some secret memory and randomness, which the adversary cannot access.

   (a) The online case: suggest an algorithm for the online case, where the goal is to discover a wrong value *as soon as it is read*. Suppose that there are N memory cells and S secret bits. What is the overhead T for each operation? Hint: there is a lower bound that $S \cdot T$ is $\Omega(n)$.

   (b) The offline problem: the goal is to discover at the end of the sequence operations whether a malfunction occurred. The transformation considered in class involved adding to each cell in memory in addition to the value stored a "timestamp", which is the time when it was (last) written. In more details, the transformation involves:

   - Scanning all memory cells at the beginning on the sequence and a writing a '0' is written in each cell.
   - After each 'read' operation a 'write' operation is performed with *current_time* as the timestamp.
   - Before each 'write' operation to location a, a 'read' operation to location a is performed.
   - At the end of the sequence of operations the memory is scanned again and all cells are read.

   Whenever a 'read' operation is performed and a timestamp t is returned, verify that $t <$ *current_time* (otherwise it is a clear malfunction of the memory). Call the property that this is always the case the **monotone property**:

Consider the following two sets:

$$R = \{(v, a, t) | \text{location } a \text{ was read with value } v \text{ and timestamp } t\}$$

$$W = \{(v, a, t) | \text{location } a \text{ was written with value } v \text{ and timestamp } t\}$$

The algorithm suggested in class tested whether these two sets are equal and that the monotone property holds. Prove the following:

Claim: If the monotone property holds, then $W = R$ iff the memory functioned properly.

3. Consider the following family of functions H where each member $h \in H$ is such that $h : \{0,1\}^\ell \mapsto \{0,1\}$. The members of H are indexed with a vector $r \in \{0,1\}^{\ell+1}$. The value $h_r(x)$ for $x \in \{0,1\}^\ell$ is defined by considering the vector $x' \in \{0,1\}^{\ell+1}$ obtained by appending 1 to x and the value is $\langle r, x' \rangle$ - the inner product of r and x' over $GF[2]$.

   Prove that the family H is three-wise independent.

4. The scenario we are considering is where a deck of n distinct cards (for simplicity labeled $1, 2, \ldots, n$) is shuffled and the cards from the deck are drawn one by one. A player called 'guesser' tries to guess the next card, for n rounds and gets a point for each correct guess. We are interested in the expected number of points the guesser can have.

   Suppose that the guesser has *perfect memory* and can recall all the cards that it has seen, then what is expected number of correct guesses?

If you have not done, please reading and watching:

- Watch the lecture by David Woodruff on "Adversarially Robust Streaming Algorithms" `https://www.youtube.com/watch?v=9qP3JCWNgnc`
- Tim Roughgarden's Notes on streaming and communication complexity `http://timroughgarden.org/w15/l/l1.pdf`