Seminar on Sublinear Time Algorithms

Lecture 5

April 21, 2010

Lecturer: Robert Krauthgamer Scribe by: Anat Ganor Updated: May 16, 2010

1 Lower Bound For Element Distinctness

Given a list of n integers, the problem of element distinctness is to determine if the list has distinct elements or is ϵ-far from it, meaning at least ϵ-fraction of the elements must be changed to make all the elements in the list distinct.
We present a lower bound on the query complexity of any algorithm that solves this problem.

Theorem 1 *Every algorithm that tests if the list has all distinct elements makes* $\Omega(\sqrt{\frac{n}{\epsilon}})$ *queries.*

Remark As a sanity check, taking $\epsilon = \frac{1}{n}$ means that there is one "interesting" element (a witness for a collision) and intuitively it requires $\Omega(n)$ queries to find it.

Proof Idea For any two specific inputs there exists an algorithm that is adapted to these inputs. For example, if we take the two inputs $x = (1, 2, ..., n)$ as a "yes" example, and $y = (1, 1, 2, 2, ..., \epsilon n, \epsilon n, \epsilon n + 1, \epsilon n + 2, ..., n(1 - \epsilon))$ as a "no" example, then an algorithm that checks the second coordinate can distinguish between them. Therefore, we look on distributions over inputs. We need two distributions, one over "yes" instances and the other over "no" instances. Then, we require that the algorithm accepts every "yes" instance and rejects every "no" instance with high probability. We assume that the algorithm is random, meaning it looks on random coordinates of the input. Given $\sqrt{n}$ samples we get $\binom{\sqrt{n}}{2} \approx n$ correlated pairs. It should find one out of ϵn pairs that are "interesting" (witnesses for collisions). The probability that a random pair is one of these witnesses is only $\frac{\epsilon n}{n^2} = \frac{\epsilon}{n}$. By looking at less than $\frac{n}{\epsilon}$ pairs we might not catch any of them. We will assume that the algorithm makes less than $\Theta(\sqrt{\frac{n}{\epsilon}})$ queries and show that statistically, it is likelt to get the same answers to its queries on inputs from D^0 and D^1, so it cannot distinguish between them with high probability. Hence, we need different distributions D^0, D^1 (the ones above are not "hard enough"). ■

Proof Attempt Consider the following distributions D^0 = uniform over all permutations of $\{1, 2, ..., n\}$ and D^1 = uniform over all permutations of $\{1, 1, 2, 2, ..., \epsilon n, \epsilon n, \epsilon n + 1, \epsilon n + 2, ..., n(1 - \epsilon)\}$. The algorithm that checks the second coordinate does not work any more. However, after only $O(\frac{1}{\epsilon})$ queries we can see a number that is bigger than $n(1 - \epsilon)$ with high probability. ■

Proof Let D^0 be uniform over all permutations of $\{1, 2, ..., n\}$ as before and D^1 be as follows: start with a uniform permutation of $\{1, 2, ..., n\}$, pick ϵn random elements from

the first half and copy them to ϵn random positions in the second half of the list. Clearly, instances given from the D^1 distribution are ϵ-far from the all distinct list.

Consider an algorithm $\mathcal{A}$ that makes q queries where $q \leq \sqrt{\frac{n}{16\epsilon}}$ and suppose for the sake of contradiction that it errs with probability at most $\frac{1}{3}$, i.e.

$$\Pr_{x\sim D^0}[\mathcal{A} \textit{ accepts } x] \geq \frac{2}{3} \textit{ and } \Pr_{x\sim D^1}[\mathcal{A} \textit{ accepts } x] \leq \frac{1}{3} \tag{1}$$

Assume for now that $\mathcal{A}$ is a non-adaptive algorithm, i.e. all queries are determined in advance, not according to answers to previous queries. In this case, we can think of $\mathcal{A}$ as an algorithm that first tosses some coins R and then proceeds as a deterministic algorithm which we denote as $\mathcal{A}_\mathcal{R}$. Let $i_1, ..., i_q \in [n]$ be the deterministic positions in the input that $\mathcal{A}_\mathcal{R}$ queries. Note that we can assume that the algorithm queries exactly q distinct positions since any other behaviour can be reduced to that.

Define $P_R^j = \Pr_{x\sim D^j}[\mathcal{A}_\mathcal{R} \text{ accepts } x]$ for $j \in \{0,1\}$. We need to show that for every R, $|P_R^0 - P_R^1|$ is small. Under D^0 the algorithm reads q random distinct values from $[n]$. Under D^1, for every pair $j \neq j' \in [q]$ it holds that $\Pr_{x\sim D^1}[x_{i_j} = x_{i_{j'}}] \leq \frac{\epsilon n}{\frac{n}{2}} \cdot \frac{1}{\frac{n}{2}} = \frac{4\epsilon}{n}$. By the union bound, $\Pr_{x\sim D^1}[\exists j \neq j' \in [q] \text{ s.t. } x_{i_j} = x_{i_{j'}}] \leq \frac{4\epsilon q^2}{n}$. If this event does not occur then $\mathcal{A}_\mathcal{R}$ sees distinct elements that are distributed uniformly, i.e. $x_{i_1}, ..., x_{i_q}$ are random q values from $[n]$. Therefore, $|P_R^0 - P_R^1| \leq \frac{4\epsilon q^2}{n} \leq \frac{1}{4}$ where the last inequality holds due to the assumption that $q \leq \sqrt{\frac{n}{16\epsilon}}$.

We saw that D^0, D^1 are "bad" for any deterministic algorithm so they are also "bad" for any distribution over deterministic algorithms, i.e. for any randomized algorithm $\mathcal{A}$ it holds that $\Pr_{x\sim D^j}[\mathcal{A} \text{ accepts } x] = E_R[P_R^j]$ and therefore,

$$\Pr_{x\sim D^0}[\mathcal{A} \text{ accepts } x] - \Pr_{x\sim D^1}[\mathcal{A} \text{ accepts } x] = E_R[P_R^0 - P_R^1] \leq \frac{1}{4} \tag{2}$$

in contradiction with (1). This is essentially based on Yao's min-max principle.

To complete the proof for adaptive algorithms, note that this property of element distinctness is invariant under permutations, so if there exists an adaptive algorithm that uses q queries then there exists also a non-adaptive one that uses the same number of queries. ■

2 Lower Bound For Testing Juntas

Definition 2 *A function $f : \{0,1\}^n \to \{0,1\}$ is called a* k-junta *if it depends on at most k of its variables, i.e. there exist $i_1, ..., i_k \in [n]$ such that $f(x) = f(y)$ whenever x, y agree on the coordinates $i_1, ..., i_k$.*

We say that f is ϵ-far from a k-junta if we need to change at least ϵ-fraction of the values of f to make it a k-junta. Note that f is given as a truth table, i.e. 2^n bits.

Here we show a result of [1].

Theorem 3 *For every $0 \leq \epsilon \leq \frac{1}{8}$, every tester for k-junta ($k < n$) must make $\Omega(k)$ queries.*

Remark

- This holds also for adaptive algorithms.
- The bound is not known to be tight. Also, it does not depend on ϵ.

Proof Let D^0 be a uniform distribution over all functions that depend only on $x_1, ..., x_{k+1}$. To create an instance of D^0 pick $g : \{0,1\}^{k+1} \to \{0,1\}$ uniformly at random and define $f(x_1, ..., x_n) = g(x_1, ..., x_{k+1})$.

Claim 4 *A function f taken from D^0 is $\frac{1}{8}$-far from k-junta (and therefore ϵ-far for any $\epsilon \leq \frac{1}{8}$) with high probability (say ≥ 0.99) for large enough k.*

Proof Idea Define random variables Z_i for every $i \in [k+1]$, indicating if there exists $x \in \{0,1\}^n$ such that $f(x) \neq f(x + e_i)$. The number of variables that f depends on is $\sum Z_i$. Use Chernoff's inequality to show that the probability that f depends on at most k variables (i.e. there exists i such that $Z_i = 0$) is smaller than $\frac{1}{100}$. ■

Let D^1 be a uniform distribution over all functions that depend only on $\{x_1, ..., x_{k+1}\} \backslash \{x_{j^\star}\}$, where $j^\star$ is chosen uniformly from $[k+1]$. Clearly, every f that is taken from D^1 is a k-junta.

Consider an (adaptive and randomized) algorithm $\mathcal{A}$ that makes $q < \frac{k}{10}$ queries. Let its queried points be $S = \{x^1, ..., x^q\}$. Note that x^i is a random variable for any $i \in [q]$ (even if the algorithm is deterministic, because they depend on $f \in D^j$ for $j \in \{0,1\}$).

Definition 5 *We say that x is a* witness *for the fact that f depends on $i \in [q]$ if $f(x) \neq f(x + e_i)$ (where e_i has only one 1 in the i^{th} coordinate).*

Definition 6 *For a given $i \in [n]$ we say that x, y are* i-twins *if they disagree only on the i^{th} coordinate, i.e. $y = x + e_i$.*

Define $\text{twins}(S) = \{i | \exists x, y \in S \text{ that are } i\text{-twins}\}$.

Claim 7 $|\text{twins}(S)| \leq |S| - 1$.

Proof Think of S as a subset of vertices in the hypercube $\{0,1\}^n$. Let E be the subset of edges such that for every i, if S contains i-twins then E contains one edge corresponding to this i-twins. Therefore, i.e. $|E| = |\text{twins}(S)|$. The graph induced by E is acyclic (because every cycle must repeat some i-twins twice), so it must be that $|E| \leq |S| - 1$. ■

Fix randomness R and call the deterministic algorithm $\mathcal{A}_\mathcal{R}$. As long as the first ℓ queries do not contain $j^\star$-twins, the distributions of $f(x^1), ..., f(x^\ell)$ under D^0 and under D^1 are the same. Meaning,

$$\Pr_{f \leftarrow D^0}[\mathcal{A}_\mathcal{R} \text{ accepts } \mid \text{didn't see } j^\star\text{-twins}] = \Pr_{f \leftarrow D^1}[\mathcal{A}_\mathcal{R} \text{ accepts } \mid \text{didn't see } j^\star\text{-twins}] \quad (3)$$

In addition, for $j \in \{0,1\}$

$$\Pr_{f \in D^j}[\mathcal{A}_{\mathcal{R}} \text{ sees } j^\star\text{-twins at query } x^\ell \mid \text{didn't see } j^\star\text{-twins before } x^\ell]$$

$$= \frac{|\text{twins}\{x^1,...,x^\ell\} \backslash \text{twins}\{x^1,...,x^{\ell-1}\}|}{k+1-|\text{twins}\{x^1,...,x^{\ell-1}\}|}$$

$$\leq \frac{|\text{twins}\{x^1,...,x^\ell\}| - |\text{twins}\{x^1,...,x^{\ell-1}\}|}{\frac{k}{2}}$$

where the last inequality holds due to Claim 7 and the fact that $\ell \leq q \leq \frac{k}{10}$.

Altogether, $\Pr_{f \in D^j}[\mathcal{A}_{\mathcal{R}} \text{ sees } j^\star\text{-twins}] \leq \frac{\frac{k}{10}}{\frac{k}{2}} = \frac{1}{5}$

$\Rightarrow \Pr_{f \in D^1}[\mathcal{A}_{\mathcal{R}} \text{ accepts } f] \leq \Pr_{f \in D^0}[\mathcal{A}_{\mathcal{R}} \text{ accepts } f] + \frac{1}{5}$

Taking expectation over R, we have $\Pr_{f \in D^j}[\mathcal{A}_{\mathcal{R}} \text{ accepts } f] = \mathbb{E}_R \Pr_{f \in D^j}[\mathcal{A}_{\mathcal{R}} \text{ accepts } f]$

$\Rightarrow \Pr_{f \in D^1}[\mathcal{A} \text{ accepts } f] \leq \Pr_{f \in D^0}[\mathcal{A} \text{ accepts } f] + \frac{1}{5}$

But the LHS is at least $\frac{2}{3}$ (since every $f \in D^1$ is a k-junta) while the RHS is at most $\frac{99}{100} \cdot \frac{1}{3} + \frac{1}{100} \cdot 1 + \frac{1}{5} < \frac{2}{3}$, and we get a contradiction. ■

References

[1] Chockler A. and Gutfreund B. A lower bound for testing juntas. *Information Processing Letters*, 2004, Volume 90, Issue 6, Pages 301-305.