Sublinear Time Algorithms 08/12/2014

Lecture 7

Lecturer: Ronitt Rubinfeld *Scribers: Boris Dogadov and Barak Gross*

1 Quick Overview And Topics To Be Covered

Lower bounds are important subject in computer science, but usually we find lower bounds for deterministic algorithms, but what about lower bounds for randomized algorithms?
In this lecture we are going to develop tools for deriving some lower bounds on randomized algorithms.
The main topics of this lecture are "Yao's Principle" and "Lower Bounds via Communication Complexity".

2 Yao's Principle

Yao's principle essentially says that given a lower bound on the expectation of some deterministic algorithm, then we can get a lower bound on the worst case expectation of randomized algorithm for the same problem.
Yao's Principle:

I there is a probability distribution D on union of YES/NO inputs such that any deterministic algorithm of query complexity $\leq t$ is incorrect w.h.p (probability $\geq \delta$ for some constant δ) for inputs according to D. Then t is a lower bound on the randomized query complexity.

We won't prove the correctness of "Yao's Principle", but if you are interested, and you have knowledge in the field of game theory, then you are more than welcome to click here for the proof.
Deterministic lower bound can be modeled by decision trees. We know already that a binary tree with m leaves needs an average of at least $\log m$ height. Now, lets apply it on the following example:

We denote by $L_n = \{w | w \text{ is a n-bit string, and } \exists u, v \ s.t. \ w = uu^R vv^R\}$.

Theorem. $\Omega(\sqrt{n})$ *queries are needed in order to test whether some input of length n belongs to* L_n.
i.e., for any tester A for this property s.t.:

- $\forall x \in L_n \; Pr(A(x) \; = \; YES) \; \geq \; \frac{2}{3}$
- $\forall x \; \epsilon - far \; from \; L_n$[1] $Pr(A(x) \; = \; NO) \; \geq \; \frac{2}{3}$

A makes at least $\Omega(\sqrt{n})$ *queries.*

Proof. w.l.o.g assume that $6|n$.
Define the following distributions:

$$N = \text{uniform distribution over all the n-bit strings that are } \epsilon - far \text{ from } L_n$$

$$P = \begin{cases} 1.\ pick\ k \in_R [\frac{n}{6}+1, \frac{n}{2}] \\ 2.\ pick\ u,v\ randomly\ s.t.\ |u| = k\ ,\ |v| = \frac{n}{2} - k \\ 3.\ output\ uu^R vv^R \end{cases}$$

$$D = \begin{cases} 1.\ Flip\ a\ fair\ coin \\ 2.\ If\ heads\ output\ according\ to\ N \\ 3.\ elseways\ output\ according\ to\ P \end{cases}$$

Define:
1. $E^-(l) = \{w \in \{0,1\}^n |\ dist(w, L_n) \geq \epsilon n \ \wedge w\ reaches\ the\ leaf\ l\}$
2. $E^+(l) = \{w \in \{0,1\}^n |\ w \in L_n \ \wedge \ w\ reaches\ the\ leaf\ l\}$
3. $F_P \ = \{\ leaf\ l\ s.t.\ A\ outputs\ YES\ at\ l\}$
4. $F_F = \{\ leaf\ l\ s.t.\ A\ outputs\ NO\ at\ l\}$

Notice that the total error of A on $D = \Sigma_{l \in F_P} Pr_{w \in D}[w \in E^-(l)] + \Sigma_{l \in F_F} Pr_{w \in D}[w \in E^+(l)]$

Claim 1. *If* $t = o(n)$ *then* $\forall l \; Pr_{w \in D}[w \in E^-(l)] \geq ((\frac{1}{2} - o(1))2^{-t}$

Claim 2. *If* $t = o(\sqrt{n})$ *then* $\forall l \; Pr_{w \in D}[w \in E^+(l)] \geq (\frac{1}{2} - o(1))2^{-t}$

Before proving the claim let's notice, that given the claims we can conclude:
Total Error Of $A = \sum_{l \in F_P} Pr_{w \in D}[w \in E^-(l)] + \sum_{l \in F_F} Pr_{w \in D}[w \in E^+(l)] \geq 2^t \cdot (\frac{1}{2} - o(1))2^{-t} \geq \frac{1}{2} - o(1) \ggg \frac{1}{3}$
Which proves our theorem.

□

[1] meaning the hamming ditance is at least ϵ

Now lets prove the claims.

Proof. Claim 1:
The idea is to Lets notice that the number of words in L_n is less or equal to the number of ways to choose index that separates the palindromes·, number of palindromes of length at most $\frac{1}{2}n$ i.e.:
$|L_n| \leq 2^{\frac{n}{2}}n$
Counting number of words that are $\epsilon - close$, is at most number of words in L_n multiplied by the ways to choose where the "errors" occurred, and how far are the strings from being in L_n, i.e.:
$\leq 2^{\frac{n}{2}}n \cdot \sum_{i=0}^{\epsilon n} \binom{n}{i} \leq 2^{\frac{n}{2}+2(\epsilon \log \frac{1}{\epsilon})\cdot n}$
For the last inequality, we used the second to the last inequality from here
We have 2^{n-t} strings that follow a path to a leaf, plus, because $t = o(n)$ and we can choose ϵ small enough such that $\epsilon \log \frac{1}{\epsilon} \leq \frac{1}{2}$ (lets say $\epsilon \lll \frac{1}{8}$), and we get that:
$E^-(l) \geq 2^{n-t} - 2^{\frac{n}{2}+2(\epsilon \log \frac{1}{\epsilon})\cdot n} > \ (1 - o(1))2^{n-t}$
So:
$Prob_{w\in D}[w \in E^-(l)] \geq \frac{1}{2}Prob_N[w \in E^-(l)] = \frac{1}{2}\frac{|E^-(l)|}{2^n} \geq \frac{1}{2}(1 - o(1))2^{-t}$
This proves claim 1.
Claim 2:
We need to show, that for every fixed set of $o(\sqrt{n})$ queries, there is a big amount of strings in L_n that path.
How many strings agree with t queries in a leaf? The answer is 2^{n-l}
How many strings in L_n agree with t queries in l? The answer is $2^{n-l}-?$
In the homework we will prove this claim.

□

3 Property Testing Lower Bounds via Communication Complexity

Communication complexity is measured in the amount of communication between processes, i.e. number of bits communicated by the processes.

Consider a 'hard' problem of **Set Disjointness** :
There are two players Alice and Bob with inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ accordingly.

$$Disj(x,y) := \bigvee_{i=1}^{n} x_i \wedge y_i$$

A known lower bound to this problem is $\Omega(n)$ bits of communication. If $\sum_{i=1}^{n} x_i \leq k$ then the lower bound is $\Omega(k)$ bits, this is knows as Sparse Set Disjontess probelm.

Define: $f : \{0,1\}^n \to \{0,1\}$ is k-linear if it is a parity of k-variables, i.e. $f(x) = x_{i_1} \oplus x_{i_2} \oplus ... \oplus x_{i_k}$, where i_j are disjoint.

Define: Property testing for k-linearity:

$$K - Linearity\ Testing = \begin{cases} if\ f\ is\ k-linear\ test\ passes\ w.p \geq 2/3 \\ if\ f\ is\ \epsilon - far\ from\ k-linear\ test\ fails\ w.p \geq 2/3 \end{cases}$$

f is $\epsilon - far$ from k-linear: $\nexists g$ such that g is k-linear $\wedge\ \{x|f(x) \neq g(x)\}| \leq 2^n\epsilon$

The reduction from set-disjointness to k-linear property testing problem :

Define
A: $f(z) := \oplus z_i$, i s.t. $x_i = 1$
B: $g(w) := \oplus w_i$, i s.t. $y_i = 1$
$h := f \oplus g$

x and y are n-bit vectors such that $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = k$, clearly f and g are k-linear.

Now we use Property-Testing algorithm to decide if h is $2k - linear$:
A runs Property-Testing algorithm on h (B runs it symmetrically to A).
Whenever A and B need the value of $g(x)$ and $f(x)$ they send each other the values, this is O(1) bits per query. Also that x is not exchanged due to the public shared randomness.

- A figures $f(x)$ out on its own
- A asks B what $g(x)$ is - B sends 1 bit
- use $h := f \oplus g$
- A sends the result to B

Both A and B use known public randomness.
Total amount of bits sent is $2 * amount\ of\ queries$.

- if $x,\ y$ are disjoint $\to\ h$ is $2k$-linear
- if $x,\ y$ are not disjoint $\to\ h$ is $1/2$-far from every $2k$-linear function ($2k-2$-linear)

One lower bound for testing set-disjointness is known, it follows that k-linearity testing has lower bound $\Omega(k)$ queries.

Example:
$x = (1000)$ $y = (0011)$ $k = 2$
$f(x) = z_{i_1} \oplus z_{i_2}$, $g(x) = z_{i_3} \oplus z_{i_4}$
$h(x) = z_{i_1} \oplus z_{i_2} \oplus z_{i_3} \oplus z_{i_4}$ - h is 4-linear

y'=(0110)
$f(x) = z_{i_1} \oplus z_{i_2}$, $g(x) = z_{i_2} \oplus z_{i_3}$
$h'(x) = z_{i_1} \oplus z_{i_2} \oplus z_{i_3} \oplus z_{i_4} = z_{i_1} \oplus z_{i_4}$ - h' is 2-linear