Sublinear Algorithms September 22, 2015

Lecture 7

Lecturer: Sofya Raskhodnikova *Scribe(s): Ishan Behoora, Ramesh Krishnan*

Today, we will see how to prove lower bounds using communication complexity. The main idea is to use known lower bounds for other models of computation and prove a reduction. This method of proving lower bounds using communication complexity was introduced by Blais et al. [1].

1 Randomized Communication Complexity model

In the randomized communication complexity model, there are two parties, Alice and Bob, wishing to compute some function. Alice has an input x and Bob has an input y. In addition, both of them have access to a shared random string. The goal is to compute a desired function $C(x, y)$, while minimizing the number of bits exchanged between the parties. The *communication complexity* of a given protocol is defined as the maximum number of bits exchanged by the protocol and the *communication complexity of a function* C, denoted $R(C)$, is the communication complexity of the best protocol for computing C.

Note that this model has no cryptographic considerations as both parties just want to compute C and this can trivially be achieved by communicating at most $min(x, y)$ bits. In addition, only the communication between the parties is charged in communication complexity model and local computation at each party is not charged.

We will use reductions to other problems with known lower bounds in Randomized Communication Complexity (RCC) model to prove lower bounds for some property testing algorithms. An important aspect of such reductions is that they are unconditional and are thus not dependent on conjectures such as $P! = NP$. We now illustrate the technique using the following example problem.

2 Set Disjointness $DISJ_k$ in RCC model

A good example of a problem in RCC model is the set disjointness problem. In this problem, both Alice and Bob are given two k element sets from the universe $[n]$ and they have to determine if their sets are disjoint. Formally, Alice has a set S where $S \subseteq [n], |S| = k$ and Bob has a set T where $T \subseteq [n], |T| = k$. They have to compute the function $DISJ_k(S, T)$, which is defined as follows:

$$DISJ_k(S,T) = \begin{cases} \textbf{accept} & \text{if } S \cap T = \phi \\ \textbf{reject} & \text{otherwise.} \end{cases}$$

The lower bound for this problem was proved by Håstad and Wigderson [3].

Theorem 1 (Håstad and Wigderson [3])**.** $R(DISJ_k) \geq \Omega(k), \forall k \leq \frac{n}{2}$.

3 Testing k-parity of boolean functions

We will use the known lower bound for Set Disjointness to prove lower bounds for testing k-parity of boolean functions. A linear functions over the finite field $\mathbb{F}_2$ is defined as follows.

Definition 2. *A Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *is* linear *(also called* parity*) if* $f(x_1, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n$ *where* $a_1, a_2, \ldots, a_n \in \{0,1\}$.

The addition operation here is modulo 2 operation (unlike the conventional addition). Note that there is no free term because the problems are equivalent for testing linearity. Linear function can also be defined in an alternate way.

Definition 3. *A linear Boolean function* $f(x_1, \ldots, x_n) = \chi_S(x) = \sum_{i \in S} x_i$ *for some* $S \subseteq [n]$.

We now define k-parity functions.

Definition 4. *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is k-parity if $f(x) = \chi_S(x) = \sum_{i \in S} x_i$ for some $S \subseteq [n]$ such that $|S| = k$, $x \in \{0,1\}^n$.*

The problem of property testing k-parity is as follows: Given a Boolean function f and an integer k as input, is the function k-parity or ϵ-far from k-parity (i.e., at least $\epsilon 2^n$ values need to be changed to make it a k-parity)? Chakraborty et al. [2] gave a tester for k-parity with running time $O(k \log k)$. Blais et al. [1] proved a lower bound of $\Omega(\min(k, n-k))$ for this problem.

We will look at the lower bound of $\Omega(k)$ for $k \leq \frac{n}{2}$. For this we will use the following.

Claim 5. *Two different linear functions $\chi_S(x)$ and $\chi_T(x)$ over $x \in \{0,1\}^n$ such that $S \neq T$ differ on half of the values of x.*

Proof. Let i be an element on which χ_S and χ_T differ. Without loss of generality, assume $i \in S \setminus T$. Pair all the n-bit strings as $(x, x^{(i)})$ where $x^{(i)}$ is x with the i^{th} bit flipped. For each such pair, it can be seen that $\chi_S(x) \neq \chi_S(x^{(i)})$, but $\chi_T(x) = \chi_T(x^{(i)})$. So, χ_S and χ_T differ on exactly one of $x, x^{(i)}$. Since all x's are paired up χ_S differs from χ_T on exactly half of the values. □

Corollary 6. *A k^*-parity function where $k^* \neq k$ is $\frac{1}{2}$-far from a k-parity function*

3.1 Reduction from $DISJ_{k/2}$ to Testing k-Parity

Now, we prove a reduction from the Disjointness problem to the problem of testing k-parity. Let T_k be the best tester for the k-parity property with $\epsilon = \frac{1}{2}$. Let q be the query complexity of T_k.

Lemma 7. *The query complexity of testing k-parity, $q = \Omega(k)$.*

Proof. We will construct a communication protocol for $DISJ_{k/2}$ that uses a reduction to T_k and has the query complexity $2q$.

Let S and T be the sets with Alice and Bob respectively such that $S, T \subseteq [n]$ and $|S| = |T| = \frac{k}{2}$. Alice computes $f(x) = \chi_S(x)$ and Bob computes $g(x) = \chi_T(x)$ for some $x \in \{0,1\}^n$. The working of the tester is public knowledge, i.e., known to both Alice and Bob. The input to the tester is generated from the shared random string. These are the steps that don't require any communication.

The communication kicks in when Alice and Bob compute $f(x)$ and $g(x)$ respectively and communicate the results to each other. Thus, 2 bits are exchanged for each query to the tester, and hence the communication complexity is $2q$.

Define function $h : \{0,1\}^n \to \{0,1\}$ as $h(x) = (f(x) + g(x)) \mod 2$. $h(x)$ is passed as input to the tester T.

Claim 8. *h is k-parity if $S \cap T = \Phi$ and k'-parity for $k' \neq k$ otherwise.*

Proof. We have

$$h = (f + g) \mod 2 = (\chi_S + \chi_T) \mod 2 = \chi_{S \Delta T}.$$

Also, $|S \Delta T| = |S| + |T| - |S \cap T|$. Thus,

$$|S \Delta T| \begin{cases} = k & \text{if } S \cap T = \Phi \\ \leq k-2 & \text{if } S \cap T \neq \Phi \end{cases}.$$

Hence h is k-parity if $S \cap T = \Phi$ and k'-parity for $k' \leq k-2$ otherwise. This completes the proof of the claim. □

By Corollary 6, any function with k' parity where $k' \neq k$ is $\frac{1}{2}$-far from k-parity function. With high probability, this function is rejected by the tester T_k. Thus S and T are disjoint if T_k accepts and not disjoint if T_k rejects.

This completes the construction of the reduction. The lower bound follows from Theorem 1.

$$2q \geq R(DISJ_{k/2}) \geq \Omega\left(\frac{k}{2}\right) \text{ for } k \leq \frac{n}{2} \text{ [By Theorem 1]}$$
$$\Rightarrow q = \Omega(k) \text{ for } k \leq \frac{n}{2}.$$

This completes the proof of the lemma. □

3.2 Notes on lower bound for adaptive testers

In the communication protocol we described, Alice and Bob communicate the results of their functions to each other, i.e., there are 2 bits of communication for each query. This is required if the tester is adaptive because the each query depends on the results of the previous queries, and hence there is an imminent need for both the parties to know the result of each query. This is the intuition behind the need for both the parties to communicate their results to each other after every query. On the other hand, if the tester is nonadaptive, then one way communication is sufficient, i.e., only Bob needs to communicate his result to Alice and Alice is not required to send her result to Bob. In this case, at the end of computation, Alice can send the result of the computation to Bob and the reduction still holds.

References

[1] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 210–220, 2011.

[2] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1683–1702, 2011.

[3] Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007.