Sublinear Algorithms March 22, 2012

Lecture 19

Lecturer: Sofya Raskhodnikova *Scribe(s): Ye Zhang*

1 Testing Monotonicity on POset Domains

In today's lecture, we consider testing monotonicity on general poset domains [FLN+02]. Specifically, we consider properties of n-bit strings:

1. Each property f is defined by a DAG G_n on n nodes;
2. Each position in the n-bit string corresponds to a node in G_n;
3. An edge (i, j) in G_n implies that $f(i) \leq f(j)$.

Testing monotonicity on general poset domains turns out to be equivalent to several other testing problems, e.g., testing properties represented by 2CNF formulas.

2 Algorithm

Now we present an algorithm to test if G_n is monotone or ϵ-far from monotone with the query complexity $\Theta(\sqrt{\frac{n}{\epsilon}})$ and running time $\Theta(\frac{n}{\epsilon})$.

Definition 1. *A pair of nodes (i, j) is called violated (by function f, with respect to G_n), if $i, j \in V(G_n)$ and there exists a path in G_n from i to j and $f(i) > f(j)$.*

For example, in **Figure** 1, $(5, 6)$ is a violated pair since $(5, 6)$ is an edge (and of course, a path) in G_n but $f(5) > f(6)$. Moreover, $(1, 3)$ is also a violated pair, as there is a path $1 \to 2 \to 3$ in G_n but $f(1) > f(3)$.

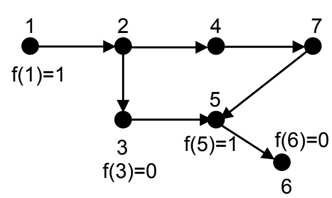

Figure 1: An example of G_7 where $V(G_7) = \{1, 2, \ldots, 7\}$

Algorithm 1: Testing Monotonicity on General Poset Domains

Input: ϵ, G_n.
1. Query $\Theta(\sqrt{\frac{n}{\epsilon}})$ values (i.e., $f(v)$ for some $v \in V(G_n)$) uniformly at random. Let S be the set of queries.
2. For each $i, j \in S$ where $i \neq j$, if (i, j) is a violated pair in G_n, **reject**.
3. Otherwise, **accept**.

The time complexity is $\binom{\Theta(\sqrt{\frac{n}{\epsilon}})}{2} = \Theta(\frac{n}{\epsilon})$.

3 Correctness

To show the correctness, we start with an observation. Recall that the transitive closure of a directed graph $G = (V, E)$ is a graph (V', E') which contains an edge $(i, j) \in E'$ iff G has a path form i to j.

Observation 2. *All graphs with the same transitive closure define the same property.*

In **Figure** 2, G_3 and $TC(G_3)$ define the same property.

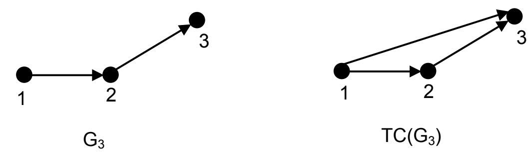

Figure 2: An example of G_3 and its transitive closure $TC(G_3)$

Therefore, w.l.o.g, we can assume that G_n is a transitive by closed graph. Now (i, j) is a violated pair in G_n if and only if (i, j) is an edge in G_n with $f(i) > f(j)$ (i.e., a violated path $\Rightarrow$ a violated edge).

First, if f is montone, Algorithm 1 will not find any violated pairs and will accept f. Now consider that f is ϵ-far from montone.

Claim 3. *If f is ϵ-far from monotone, there is a matching of size at least $\frac{\epsilon n}{2}$ violated edges.*

Proof. Consider the graph of violated edges. We claim that the distance of f from a monotone function is equal to the size of the minimum vertex cover (VC) of the graph. Note that it is sufficient to change f on nodes in a VC to make it monotone, which is a corollary from the following exercise.

Exercise 19.1. *Given a partial function (that is defined on some nodes) which violates no edges in $TC(G_n)$ (the transitive closure of G_n), we can define the values on the remaining nodes, so that the resulting function f' is monotone. [Hint: prove it by induction, defining one value at a time without introducing new violations.]*

On the other hand, it is also necessary to change f on nodes in a minimum VC. Otherwise, some violated edges remain.

The fact that f is ϵ-far from monotone implies that there is a minimum VC of size ϵn. Using the fact that the size of maximum matching $\geq \frac{|\min VC|}{2}$, we have a matching of size $\geq \frac{\epsilon n}{2}$, which completes the proof.

□

Exercise 19.2. *Prove that if a Boolean function f is ϵ-far from monotone, there is a matching of $\geq \epsilon n$ violated edges.*

So, if f is ϵ-far from monotone, the matching of size at least $\frac{\epsilon n}{2}$ violated edges induces a bipartite graph in G_n. We call those violated edges witness edges. One side endpoints in the witness edges are called top witnesses and the other side endpoints are called bottom witnesses.

Now, let F_T be the event that we see $\leq \frac{\epsilon s}{4}$ top witnesses in the s samples. Similarly, we define F_B be the event that we see $\leq \frac{\epsilon s}{4}$ bottom witnesses in the s samples. Moreover, we define F to be the event that no witness edge is detected. We want $\Pr[F]$ to be small. To bound it, we have

$$\Pr[F] \leq \Pr[F_T] + \Pr[F_B] + \Pr[F | \overline{F_T} \wedge \overline{F_B}].$$

We know that $\Pr[F_T]$ and $\Pr[F_B]$ are both bounded by small probabilities (following Chernoff-Hoefding inequality). Specifically, let $X_i = \begin{cases} 1 & \text{if } i\text{-th sample is a top witness;} \\ 0 & \text{otherwise.} \end{cases}$. Let $X = \sum_{i=1}^{s} X_i$. $\mathbb{E}(X) = \sum_{i=1}^{s} \Pr[X_i = 1] \geq s\frac{\epsilon n/2}{n} = \frac{\epsilon s}{2}$. $\Pr[F_T] = \Pr[X \leq \frac{\epsilon s}{4}] < e^{-\frac{\epsilon s}{16}}$.

On the other hand, see **Figure** 3,

$$\Pr[F|\overline{F_T} \wedge \overline{F_B}] \leq (1 - \frac{\epsilon s/4}{\epsilon n/2})^{\frac{\epsilon s}{4}} \leq e^{-\frac{s^2}{n}} \leq \frac{1}{3}$$

when $s = \Theta(\sqrt{\frac{n}{\epsilon}})$.

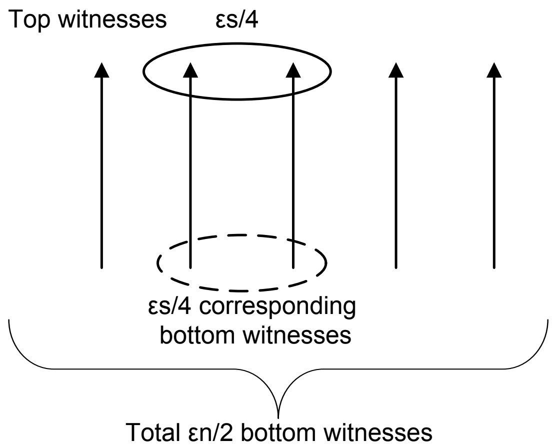

Figure 3: Analysis of $\Pr[F|\overline{F_T} \wedge \overline{F_B}]$

4 Lower Bound on Query Complexity

Theorem 4. *Every non-adaptive (two-sided) ϵ-tester for monotonicity of graph labellings must query $n^{\Omega(\frac{1}{\log\log n})}$ values.*

By Yao's Minimax Principle, to prove a lower bound q on query complexity of randomized algorithms, it is enough to show two distributions P and N on input such that it is hard for q-query deterministic algorithms to distinguish P from N. Distribution P is on positive instances and distribution N is on negative instances.

Specifically, to define "hard to distinguish", we consider queries made by a q-query tester where $a_1, a_2, \ldots, a_q$ are the answers of the q queries.

Definition 5 (P-view)**.** *Let P be a distribution on inputs. P-view denotes the distribution on the answers $(a_1(X), a_2(X), \ldots, a_q(X))$ where $X \sim P$.*

To define "hard to distinguish", we use the statistical distance.

Definition 6 (Statistical Distance)**.** *Let D_1, D_2 be two distributions. The statistical distance between D_1 and D_2 is:*

$$SD(D_1, D_2) = \max_{S \subset support(D_1) \cup support(D_2)} |\Pr_{x \leftarrow D_1}[x \in S] - \Pr_{x \leftarrow D_2}[x \in S]|.$$

Now we require that $SD(P\text{-view}, N\text{-view}) \leq \frac{1}{3}$.

References

[FLN+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *STOC*, 2002.