6.896 Sublinear Time Algorithms March 15, 2007

Lecture 12

Lecturer: Ronitt Rubinfeld *Scribe: Jeremy Hurwitz*

1 Intro to Property Testers

1.1 Informal Overview

So far, we've dealt with two main types of approximations - multiplicative and additive. Such approximations work well for optimization problems and problems whose output is over an interval of reals, such as the distance between two distributions or the entropy of a distribution. Unfortunately, such a definition of "approximately" fails for decision problems. For example, we may want to ask "Is G connected?" or "Is G bipartite?" How, then, do we approximate a yes/no answer?

Instead of approximating the answer, we will approximate the input. "Is G connected?" becomes "Is G almost a connected graph?" Such questions are often difficult, but in some cases we can answer such questions in sublinear time.

A property tester is an algorithm that answers this question. The tester outputs "yes" with high probability if the input has the desired property and answers "no" with high probability if the input is ϵ-far from having the property.

1.2 Formal Definition

Given a property P and a domain D, let $\mathcal{P} = \{x \in D \mid x \text{ has property P}\}$.

Definition 1 (Decision Algorithm) *A decision algorithm $\mathcal{A}$ is defined by*

$$\begin{aligned}
&\text{if } x \in \mathcal{P},\ \mathcal{A}(x) = pass \\
&\text{if } x \notin \mathcal{P},\ \mathcal{A}(x) = fail
\end{aligned}$$

Definition 2 (ϵ-far from $\mathcal{P}$) *Given a metric $d(x, y)$ on D, let $d(x, \mathcal{P}) = \min_{y \in \mathcal{P}} d(x, y)$. x is ϵ-far from $\mathcal{P}$ if $d(x, \mathcal{P}) > \epsilon$.*

Definition 3 (Property Tester) *A property tester $\mathcal{T}$ is defined by*

$$\begin{aligned}
&\text{if } x \in \mathcal{P},\text{, then with high probability } \mathcal{A}(x) = pass \\
&\text{if } x \text{ is } \epsilon\text{-far from } \mathcal{P}\text{, then with high probability } \mathcal{A}(x) = fail
\end{aligned}$$

Notice that if $0 < d(x, \mathcal{P}) \leq \epsilon$, $\mathcal{A}$ is undefined.

As indicated by the input to $\mathcal{T}$, we must always specify the metric being used by the property tester. We must also specify the representation of the inputs and what access methods we are allowed. These two choices can have a large influence on the running time of the algorithm.

Common distance functions for graphs with n vertices include

1. the number of edges that must be inserted or deleted to turn x into y
2. the number of edges that must be inserted or deleted, divided by n
3. the number of edges that must be inserted or deleted, divided by n^2

For strings, the metric might be the Hamming distance or edit distance.

Common representations of graphs are adjacency lists, as we saw in lecture 11, and adjacency matrices, which will be defined below.

Definition 4 (1-sided Error Tester) *A **1-sided error tester** is a property tester that always outputs pass if $x \in \mathcal{P}$*

Placing additional requirements on a property tester, such as 1-sided error, often make it much easier to prove lower bounds.

Definition 5 (Adjacency Matrix Model) *Given a graph $G = (V, E)$, an algorithm in the Adjacency Matrix Model receives G as input in the form of a matrix A such that*

$$A_{ij} = \begin{cases} 1 & (i, j) \in E \\ 0 & (i, j) \notin E \end{cases}$$

When using the adjacency matrix model, most property testers use the third metric given above, in which the number of altered edges is divided by n^2.

2 Bipartite Tester

Definition 6 (Bipartite Graph) *Given a graph $G = (V, E)$, G is **bipartite** if there exists a partition (V_1, V_2) of V such that for all $(u, v) \in E$, u and v do not lie in the same partition.*

Definition 7 (Violates) *Given a partition (V_1, V_2) of V and an edge $(u, v) \in E$, we say that (u, v) **violates** (V_1, V_2) (or (u, v) is a **violation** of the partition) if u and v lie in the same V_i. So G is bipartite if and only if there exists a partition of V with no violations.*

Since adding an edge can never turn a non-bipartite graph into a bipartite graph, we have the following definition for distance for bipartite graphs.

Definition 8 (ϵ-far from bipartite) *G is ϵ-far from bipartite if we must remove $> \epsilon n^2$ edges to make it bipartite.*

Given a graph G, we can determine if the graph is bipartite in time linear in the number of edges, by using breadth-first-search. Starting at any node, we mark that node with 1. Each of its neighbors is marked with 2. Each of their neighbors is marked with 1, and so on. If at any point we get contradictory assignments at a node, then the graph is not bipartite. Otherwise, it is.

We are now ready to give a bipartite tester.

BIPARTITE(G, ϵ)

1. Pick $\theta(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$ nodes uniformly. Call this sample S.
2. Query all edges between nodes in S. At this point, we have looked at a $|S| \times |S|$ submatrix.
3. Check if the known submatrix, $S \times S$, is bipartite, and return the result.

Since checking bipartiteness takes linear time in the size of the matrix, the query complexity and running time of the algorithm is $O(\frac{1}{\epsilon^4} \log^2 \frac{1}{\epsilon})$.

Remark In the degree-bounded model, using the ϵn metric instead of ϵn^2, the sample complexity is $\Omega(\sqrt{n})$.

Clearly, if G is bipartite, the algorithm will always return the correct answer. So, assuming the algorithm works in the negative case, we have that the algorithm is actually a 1-sided tester.

We will now attempt to prove the negative case. While the proof will turn out to be faulty, it will give some insight as to what is happening.

Proof Attempt First, we give an equivalent definition of being ϵ-far from bipartite.

Observation 9 *$G = (V, E)$ is ϵ-far from bipartite if and only if for all partitions (V_1, V_2) of V, then E contains at least ϵn^2 violations of the partition.*

Consider any partition. Since G is ϵ-far from bipartite, there must be at least ϵn^2 violations. Taking a sample of size $m = \theta(\frac{1}{\epsilon} \log \frac{1}{8})$, we have (at least) $m/2$ independant pairs. Therefore, the sample hits some violating edge with probability greater than $1 - (1-\epsilon)^{m/2} = 1 - \delta$. ■

Problem with the Proof Attempt

- We need to rule out all partitions, not just one. We would therefore need to have $\delta < 2^{-n}$ so that we could use a union-bound.

To fix this problems, we will instead show that

$$S \times S \text{ bipartite } \Rightarrow G \text{ is close to bipartite}$$