Introduction to Property Testing
Lecture 2

Lecture by Dr. Reut Levi
Typeset by Steven Karas

2020-04-22
Last edited 18:23:23 2020-04-22

Disclaimer These notes are based on the lectures for the course Introduction to Property Testing, taught by Dr. Reut Levi at IDC Herzliyah in the spring semester of 2019/2020. Sections may be based on the lecture slides prepared by Dr. Reut Levi.

1 Agenda

- Recap
- Additional examples of property testing

2 Binary Sequences

Note that in all the examples today we will be using binary sequences as our domain with relative Hamming distance as our distance metric.

Let $x, z \in \{0,1\}^*$ be binary sequences. Let $\delta(x,z) = \begin{cases} |\{i \in [1x1] : x_i \neq z_i\}| & \text{if } |x| = |z| \\ \infty & \text{otherwise} \end{cases}$

Let S be some set of binary sequences. We define the distance of x from S as:

$$\delta_S(x) = \min_{z \in S}\{\delta(x,z)\}$$

We define distinguishing sequences more distant than some ε for randomized algorithms as if $x \in S$ then the algorithms accepts with probability $> \frac{1}{2}$, and if x is more than ε far from S then rejects with probability $> \frac{1}{2}$[1]

2.1 Majority

We define the property of majority as the set of all sequences which have strictly more than half of their bits as 1.

$$\text{MAJ} = \left\{ x : \sum_{i=1}^{|x|} x_i > \frac{|x|}{2} \right\}$$

2.1.1 Lower bound on number of queries

Last time we proved this for any A that makes $< \frac{n}{3}$ queries. We claim that if $X_n \sim D_1$ and $Y_n \sim D_2$ then

$$|\Pr[A(X_n) = 1] - \Pr[A(Y_n) = 1]| \leq \frac{q}{n}$$

where A is an algorithm that makes q queries.

[1] I use $\frac{1}{2}$ instead of the $\frac{2}{3}$ used in the slides

Proof As a sketch, we think of a random process that answers queries on the fly as such:

To determine X_ns, first pick i u.a.r.[2] from $[n]$ as the special bit that we fix to be zero. Then pick N u.a.r. from strings with Hamming weight $\lfloor \frac{n}{2} \rfloor$ with a 0 in location i.

To determine Y_ns, first pick i u.a.r. from $[n]$ as the special bit that we fix to be one. Then pick Y u.a.r. from strings with Hamming weight $\lfloor \frac{n}{2} \rfloor$ with a 0 in location i and then flip that bit.

The random process whenever queried on a "new" bit, it flips a coin to decide if this bit is special. The other bits are chosen to be 1 or 0 on the fly as well. Observe that if the random process does not pick a special bit then its answers are the same regardless of if it is a NO or YES instance process.

The probability to pick a special bit if we make q queries is therefore $\frac{q}{n}$ from the union bound.

2.1.2 Limitations of deterministic algorithms

Any deterministic algorithm that distinguishes between MAJ and 0.5-far from MAJ must make $\frac{n}{2}$ queries.

Proof Assume there exists an algoirthm A that makes $q < \frac{n}{2}$ queries. Consider the execution of A such that each query is answered with 0. This execution is consistent with the all-0 string and a string with Hamming weight $n - q$ ("1"s everywhere else).

2.2 Symmetric properties

We define a symmetric property as for any $x \in \{0,1\}^*$ and every permutation over $[|x|]$, $x \in S$ iff $X_{\Pi(1)}, \ldots, X_{\Pi(k)} \in S$ then S is symmetric.

For any symmetric property of binary sequences S, there exists a randomized algorithm that makes $O(\frac{1}{\varepsilon^2})$ queries and distinguishes between $x \in S$ and x that are ε far from S. Note that we can guarantee this for any fixed alphabet, and this does generalize for an unbounded alphabet (exercise 1.3).

Proof Note that we can decide if $x \in S$ or not just by knowing $wt(x)$. More formally:

$\forall n \exists W_n \subseteq [n]$ such that $\forall x \in \{0,1\}^n$

2.3 Beyond symmetric properties

2.3.1 Sorted

A sequence $x \in \{0,1\}^*$ is sorted iff $x_i \leq x_{i+1}$ $\forall i \in [|x| - 1]$.

There exists a randomized $O(\frac{1}{\varepsilon})$ time algorithm for ε-testing if a sequence is sorted.

In other words, the sequence is all 0s followed by all 1s (may be missing 0s or 1s)

Algorithm On input $x \in \{0,1\}^n$, query x in locations $\frac{\varepsilon n}{2} \cdot i$ for every $i \in [\frac{2}{\varepsilon}]$. Then query x at $m = O(\frac{1}{\varepsilon})$ locations u.a.r. Accept iff the induced substring is sorted.

Correctness If substring induced by step 1 is not sorted, then we reject. If it is sorted and $x \in S$ then we can determine x up to a single interval; specifically the interval where we switch from 0s to 1s. If x is ε-far from SORTED, then there exists $\geq \frac{\varepsilon n}{2}$ indexes such that we can reject if we sample those indexes.

[2] uniformly at random

3 Formal definitions

- Properties of function which represent obejcts
- prop e by querying the function
- n - size of domain, w.l.o.g. domain $[n]$
- Range R_n which may depend on n
- property: set Π_n of functions from $[n]$ to R_n
- For the sake of asymptotic analysis and algorithmic uniformity we let n vary and consider testers for $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$.

for example, this means that some algorithms will branch if n is even or odd

Input is oracle access to $f : [n] \to R_n$ with proximity parameters $\varepsilon > 0$ and n.
Distance is defined for two functions $f, g : [n] \to R_n$.

3.1 Property Tester

A tester for property Π is a probabilitistic algorithm denoted T that receives as input query access to $f : [n] \to R_n$ and parameters n and ε such that:

1. if $f \in \Pi_n$ then $\Pr[T(f, n, \varepsilon) = 1] \geq \frac{2}{3}$
2. if $\delta_\Pi(f) > \varepsilon$ then $\Pr[T(f, n, \varepsilon) = 0] \geq \frac{2}{3}$

If $\Pr[T(f, n, \varepsilon) = 1] = 1$ for every $f \in \Pi_n$ then T has one sided error. Otherwise, we say T has two sided error.

A tester has query complexity $q : \mathbb{N} \times (0, 1] \to \mathbb{N}$ if on input n, ε and oracle access to any $f : [n] \to R_n$ the tester makes at most $q(n, \varepsilon)$ queries. Note that in this field we want algorithms that are sublinear to n.

3.2 Nonadaptivity

The queries of an algorithm are determined by n, ε, not the results of the query.

4 Homework

She suggested writing extremely formal and rigorous solutions. From the book she wants us to do exercise 1.1 and exercise 1.3 due two weeks from now (this will be added in moodle)

5 Note regarding missing sections in these notes

The lecture notes were scrolled a little too fast for me to scribe them in real time, so I will take the effort to update these notes in a few days.

6 For next time

Due to independence day, the next lecture will take place in two weeks.

References

[1] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.