UNIVERSITÄT DES SAARLANDES

max planck institut informatik

Sublinear Algorithms

Lecture 07: Property Testing I

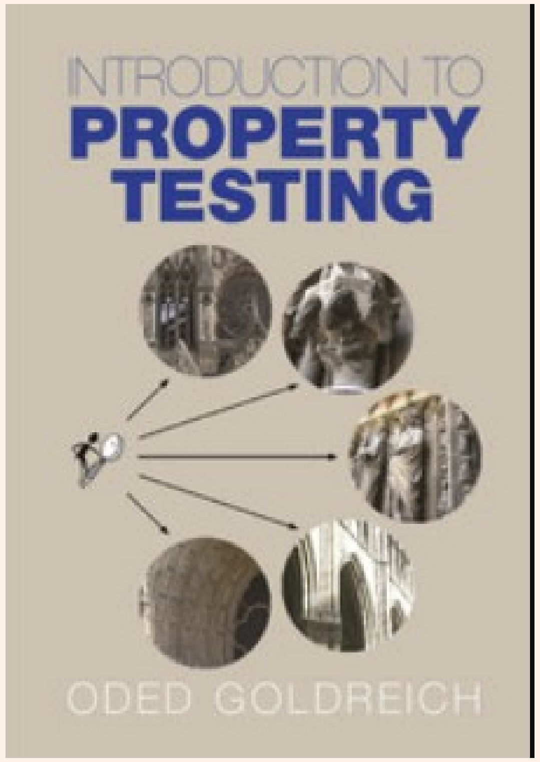
European Research Council
Established by the European Commission

Previous 3 lectures

Measurements

Next 3 lectures

Property Testing



Property Testing

Goal: Infer a property of an enormous object by looking at a small fraction of it

Difference from sparse recovery:
In sparse recovery you **compress** the whole object
Via measurements

Minimize measurements
Vs
Minimize Accesses to the object and **Time**

Form of Property Testing

Decide whether an object **has** a property

Or

is **far** from having it

Is a graph G on m edges bipartite, **or**

one needs to remove more than εm edges to make it bipartite?

Are two sequnces of length n equal, **or**

does one need to delete at least εn characters from each to make them equal?

Is a graph triangle free, **or**

Does one need to delete at least εm edges to make it triangle free?

Computational Cost

Given : **Oracle access** to the object
Minimize #oracle accesses + **running time**

Example 1. **Oracle access** to a graph in the following way:
For vertices u,v, are u and v connected?

Example 2. **Oracle access** to two sequences in the following way:
Given a position i, do they have the same symbol on i?

Formal definition

A property testing algorithm
for a decision problem L
with
query complexity $Q(n)$,
time complexity $T(n)$,
proximity parameter ε,

Is a randomized algorithm
which on input x
makes $Q(|x|)$ queries, runs in $T(|x|)$ time

If x belongs to L, accepts wp 0.9
If x is ε-far from L, rejects wp 0.9

Testing Monotonicity

You may think of it as an n-length array

$$f : [n] \to R$$

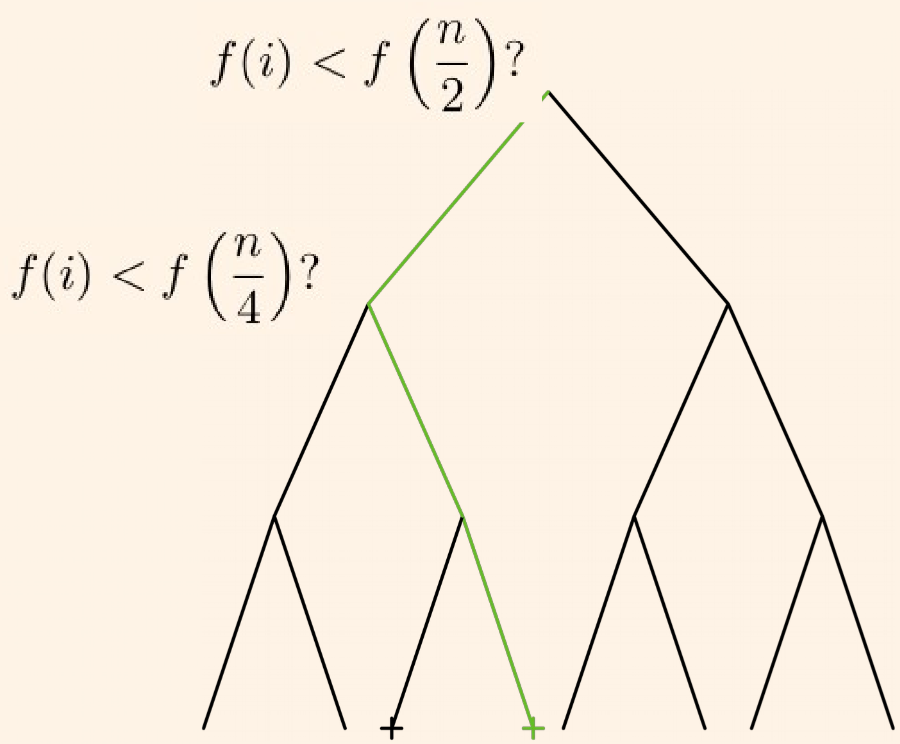
Totally ordered set

f is monotone if $\forall x, y$ with $x < y : f(x) < f(y)$

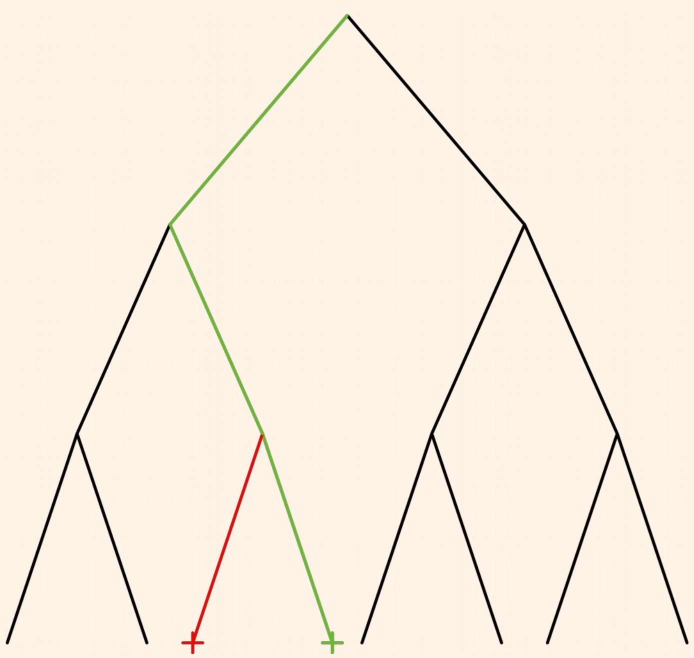
Decide whether f is δ-far from monotone

Pick random $i \in [n]$

Run the standard executation of binary search on f, and test whether you arrived at i.



Binary search arrived at i



Binary search did not arrive at i

$1 + \lceil \log_2 n \rceil$ queries

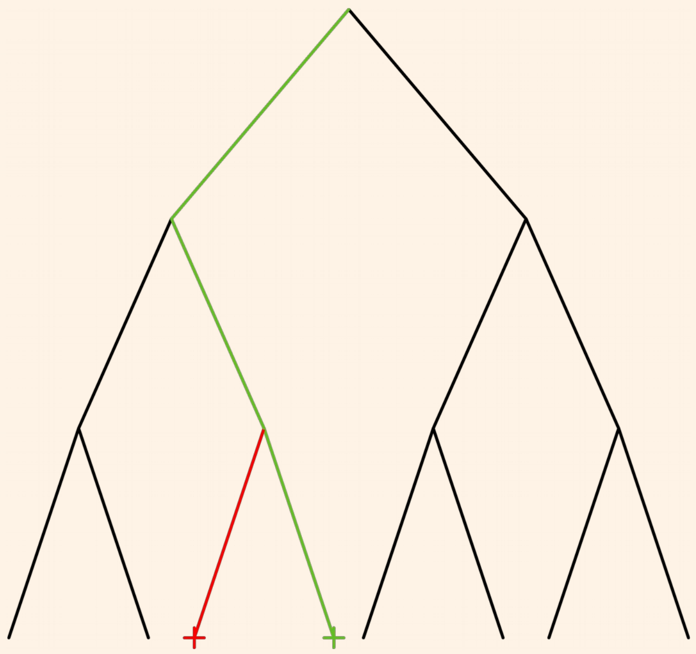
Testing Monotonicity

Randomness only over the choice of i

Call i good if the execution ends on i

Claim I: If both $i<j$ are good then $f(i) < f(j)$

$t \in [\lceil \log_2 n \rceil]$:point where first step in which the binary searches for i,j split



$f(i) < f(t) < f(j)$

■

Claim II: The restriction of f on good points yields a monotone sequence

$i_1, i_2, \ldots, i_G$ are good $\qquad f(i_1) < f(i_2) < \ldots < f(i_G)$ ■

Probability of acceptance at least G/n

Analyzing the binary search algorithm

If prob of acceptance is at least 1-δ
then
f is δ-close to monotone

Becuse *G* is at least *(1-δ)n,*
δn substitutions suffice

f is *δ*-far from monotone, then prob of acceptance < 1-δ

How many times do I need to run the tester
to decide if probability of acceptance is 1 or < 1-δ?

Equivalent: How many times to I need to throw a coin
to decide whether heads happens wp 1 or <1-δ?
(problem set)

Can distinguish between monotone
or *δ*-far from monotone

Testing Linearity

Important:
Over {0,1}^n
Addition is the same as
Subtraction

A {0,1} *n*-dimensional vector

$$f : \{0,1\}^n \to \{0,1\}$$

f is linear if $\forall x, y : f(x+y) = f(x) + f(y)$

$$x = (x_1, x_2, \ldots, x_n)$$

$$\exists (a_1, a_2, \ldots, a_n) \in \{0,1\}^n : f(x) = \left(\sum_{i=1}^{n} a_i x_i \right) \bmod 2$$

Blum-Luby-Rubinfield test

Pick 2 points *x,y* and test whether *f(x) + f(y)= f(x+y)*

Soundness of the BLR test:

If *f is δ-far* from linear, then

$$\Pr\{\text{BLR rejects } f\} \geq \min\left\{\frac{2}{9}, \frac{\delta}{2}\right\} \geq \frac{2\delta}{9}$$

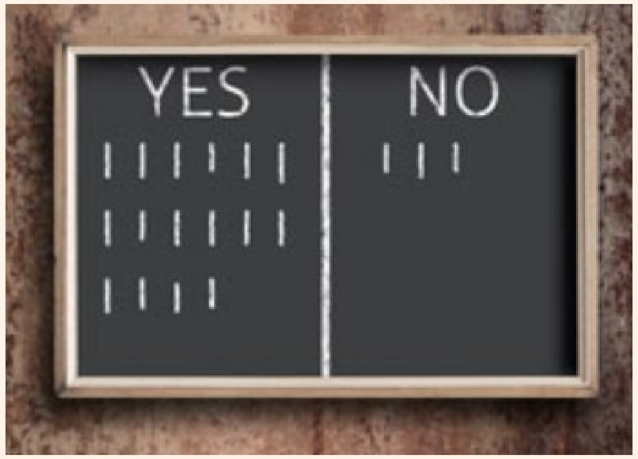
Pick *Ω(1/δ)* pairs to ensure rejection

Analysis of the BLR Test

$$g(x) = 1, \text{if } \Pr\{f(y) + f(x-y) = 1\} \geq \frac{1}{2}$$

$$g(x) = 0, \text{otherwise}$$

Majority vote



$$g(x) \neq f(x) \rightarrow_{\text{at least half of the y}} : f(y) + f(y-x) \neq f(x) \longrightarrow f(x+y) \neq f(x) + f(y)$$

$$\Pr\{rejection\} \geq \Pr\{f(x) \neq g(x)\} \cdot \Pr\{\text{bad y for x is chosen}\} \geq \text{dist}(f,g) \cdot \frac{1}{2}$$

$$\text{dist}(f,g) = |\{x \in \{0,1\}^n : f(x) \neq g(x)\}|$$

Structural Claim: If probability of rejection < 2/9 then g is linear

Conclusion:
Either rejection probability is constant, or
is at least Ω(dist(*f*,*linear*)) = Ω(δ)

Proof of the Structural Claim

Structural Claim: If probability of rejection < 2/9 then g is linear

$$g(x) = 1, \text{if } \Pr\{f(y) + f(x-y) = 1\} \geq \frac{1}{2}$$

$$g(x) = 0, \text{otherwise}$$

$$B_x = |y \in \{0,1\}^n : g(x) = f(y) + f(x-y)|$$

$$B_x \geq \frac{2^n}{2} = 2^{n-1}$$

Intermediate Structural Claim: If probability of rejection < 2/9 then $\forall x, B_x > \frac{2}{3} \cdot 2^n$

Assuming intermeditate claim: Pick random z.
Each one of the following holds with >2/3 probability.

1. $f(z) + f(x+z) = g(x)$
2. $f(z) + f(y+z) = g(y)$
3. $f(z+x) + f(z+y) = g(x+y)$

And all simultaneously with positive probability

so add them up... g is linear!

Proof of the Intermediate Structural Claim

Intermediate Structural Claim: If probability of rejection < 2/9 then $\forall x, B_x > \frac{2}{3} \cdot 2^n$

$$g(x) = 1, \text{if } \Pr\{f(y) + f(x-y) = 1\} \geq \frac{1}{2}$$

$$g(x) = 0, \text{otherwise}$$

$$B_x = |y \in \{0,1\}^n : g(x) = f(y) + f(x-y)|$$

$$B_x \geq \frac{2^n}{2} = 2^{n-1}$$

Fix x, and double count:

$$V = |\{(y,z) : f(y) + f(x+y) = f(z) + f(x+z)\}|$$

Claim I: $$V = \underbrace{B_x^2}_{\text{both terms equal to } g(x)} + \underbrace{(2^n - B_x)^2}_{\text{both terms equal to } g(x)+1}$$

Claim II: $$V > \frac{5}{9} \cdot 4^n$$

Combine claims for a lower bound

Sketch of proof of Claim II: Condition in V equivalent to

$$f(y) + f(z) = f(x+y) + f(x+z)$$

Satisfied for at least 5/9 fraction of pairs *(y.z) by assumption on BLR*

Recap

Structural Part + **Algorithmic** Part

Often simple algorithms,
and all the complexity is pushed to analysis

Structural part: Understand how an object
which is δ-far from having a property looks like

Thank you