Lecture 11:

Monotonicity Testing
Property Testers For Monotonicity:

Given: list \( y_1, \ldots, y_n \)

Output: sorted?

i.e. if \( y_1 \leq y_2 \leq \ldots \leq y_n \) output PASS (with prob \( \geq \frac{3}{4} \))

if \( y_1, \ldots, y_n \) \( \varepsilon \)-far from sorted (need to delete \( E_n \)?)

Output: FAIL (with prob \( \geq \frac{3}{4} \))

E.g.

Sorted: 1 2 4 5 7 11 14 19 20 21 23

Close: 1 4 2 5 7 11 14 19 20 39 23

Far: 40 39 23 38 4 5 21 20 19 2

An easy case: \( y_i \in \{0,1,3\} \forall i \)

Can do it in \( \text{poly}(\frac{1}{\varepsilon}) \) time. (hw.)
A first attempt:

Proposed algorithm: "neighbor test"
Pick random \( i \), test \( y_i \leq y_{i+1} \)

Bad input:
\[ 1, 2, 3, 4, 5, \ldots, \frac{n}{4}, 1, 2, 3, 4, \ldots, \frac{n}{4}, \frac{1}{2}, 3, 4, \ldots, \frac{n}{4} \]
- \( \frac{3}{4} n \)-far from monotone
- only 3 choices of \( i \) fail

A second attempt:

Proposed algorithm: "random pair test"
Pick random \( i < j \), test \( y_i \leq y_j \)

Bad input: \( \frac{n}{4} \) groups of 4 decreasing elements

\[ 4, 3, 2, 1, 7, 6, 5, 12, 11, 10, 9, 16, 15, 14, 13, \ldots \]
- largest monotone sequence size \( \frac{n}{4} \)
- must pick \( i, j \) in same group to fail, \( \text{prob} \leq \frac{1}{n} \)
  if see \( o(\frac{1}{n}) \) samples, \( \text{prob} \ o(1) \)
A minor simplification:

Let's assume list is distinct

**Claim** This is wlog

why? (old trick used in parallel computation)

\[ X_1, \ldots, X_n \Rightarrow (X_1, 1), (X_2, 2), \ldots, (X_n, n) \]

"virtually" (at runtime)
append i to each \( X_i \)

breaks ties w/o changing order

i.e. if \( X_i \leq X_{i+1} \) then \( (X_i, i) \leq (X_{i+1}, i+1) \)

A test:

Given \( X_1, \ldots, X_n \)

Repeat \( O(\varepsilon) \) times:

Pick \( i \in \mathbb{R}[n] \)

\( Z \leftarrow X_i \)

do binary search on \( X_1, \ldots, X_n \) for \( Z \)

if see any inconsistency \( \text{FAIL} \) \( \text{halt} \)

i.e. left is bigger
right is smaller

if end up at locn \( j \neq i \) \( \text{FAIL} \) \( \text{halt} \)

Pass

runtime \( O(\log \frac{n}{\varepsilon}) \)
Why does it work?

- If \( x_1 < x_2 < \ldots < x_n \) then always passes.

- To show: if need to change \( z \in E \) \( x_i \)'s then test fails why equivalently: if test likely to pass, \( x_i \)'s \( \varepsilon \)-close to monotone
defn. i "good" if bin search for \( z \leq x_i \) successful

Reformulation of test:
- Pick \( O(\varepsilon) \) \( x_i \)'s randomly & pass if all are good

if test likely to pass, \( \geq 1 - \varepsilon \) fraction of \( x_i \)'s are good
(Otherwise, in \( O(\varepsilon) \) samples, likely to hit a bad \( x_i \))

Main observation:
- "good" elements form increasing subsequence

Proof: if \( i < j \) both good, let \( K \) be least common ancestor in bin search tree.

When hit \( x_K \), search for \( x_i \) went left & search for \( x_j \) went right.

so \( x_i < x_K < x_j \)
Monotonicity over Posets:

def f is monotone over poset P if:

∀ x ≤ y
then f(x) ≤ f(y)

Examples: Can represent via daggs

- bipartite posets

- hypercube

In h.w.: Show testing mononicity of arbitrary poset can be transformed into "equivalent" monotonicity testing problem on bipartite poset
If can test monotonicity, can also test:

1) Given $2CMF \Phi$ along with assignment $A = \{a_1, ..., a_n\}$, $a_i \in \{T, F\}$
   - Pass if $\Phi(A) = T$
   - Fail if $\forall A' s.t. A \epsilon$-close to $A'$, $\Phi(A) = F$ \(\Rightarrow\) whp

2) Given $G$ with $U \subseteq V$
   - Pass if $U$ is VC
   - Fail if $\forall U' s.t. U \epsilon$-close to $U'$, $U'$ not VC

3) Given $G$ with $U \subseteq V$
   - Pass if $U$ is clique
   - Fail if $\forall U' s.t. U \epsilon$-close to $U'$, $U'$ not clique

Then for bipartite graphs ($n$ nodes on each side)
$\epsilon$-mon test can be done in $O(\sqrt{n\epsilon})$ queries

PT: h.w.

Then $\epsilon$-mon test requires $n^{\omega(1)}$ queries if nonadaptive?

\(\Rightarrow\) $\Omega(\log n)$ queries adaptive

Can we improve this to $\Omega(\sqrt{n})$?
For adaptive queries?
What about grids?

\( f: [n] \times [n] \to [m] \)

Can test monotonicity in \( O(\log^2 n) \) time

\( f: [n]^d \to [m] \)

Can test monotonicity in \( O\left(\frac{d}{\varepsilon} \log n \log m\right) \)

\( f: 2^d \to \{0, 1\}^d \)

Can test monotonicity in \( O\left(\frac{d^2 \log d}{\text{poly}(\varepsilon)}\right) \)

need \( \Omega\left(\frac{d^{1/4}}{\sqrt{n}}\right) \) queries (even for adaptive algorithms!)
Monotonicity in $L_1$ distance

Problem:
- given $f : [n] \rightarrow [0,1]$

Pass if $f$ is monotone
Fail if $f$ is $\epsilon$-far in $L_1$-distance from any monotone fcn.
(i.e. $L_1$-dist is $\geq \frac{\epsilon}{n}$)

How does it compare to Hamming distance?

when $f : [n] \rightarrow \{0,1\}$, Hamming distance equals $L_1$ distance
for $f : [n] \rightarrow \{0,1\}$, $HD = L_1$-dist
for $f : [n] \rightarrow [0,1]^d$, $HD \cdot d \geq L_1$-dist

Thm: Can test if $f$ monotone with respect to $L_1$-distance in $O(\frac{1}{\epsilon})$ queries

Proof: idea: reduce to Boolean fcn monotonicity testing!

def. for $\epsilon \in [0,1]$, threshold fcn $f_{(\epsilon)} : [n] \rightarrow \{0,1\}$

$$f_{(\epsilon)}(x) = \begin{cases} 1 & \text{if } f(x) \geq \epsilon \\ 0 & \text{otherwise} \end{cases}$$
Express $f$ as sum of funcns mapping to $\mathbb{R}_0^+$:

For any func $f$ (including non-monotone) s.t. $f(0) = 0$:

$$f(x) = \int_0^1 f(t) \, dt = \int_0^1 f_{(x)} \, dt$$

$$= \int_0^1 f(x) \, dt + \int_0^1 f_{(x)} \, dt$$

= 1 in this range

= 0

Let $L_1(f, M) = L_1$ distance of $f$ to closest monotone func.

Lemma $L_1(f, M) = \int_0^1 L_1(f_{(t)}, M) \, dt$

ie. can express dist to monotonicity in terms of distances of threshold funcns.

pf idea

To change $f$ into monotone

must make each "row" monotone. (row = 0,1 func from choice of $t$)

need to change at least $L_1(f_{(0)}, M)$ in each row

$$\Rightarrow L_1(f, M) \geq \int_0^1 L_1(f_{(t)}, M) \, dt$$

When make each row monotone, let $g_t$ be resulting func

st. $g_t$ is closest monotone func to $f_{(t)}$ & let $g = \int_0^1 g_t \, dt$

$g$ monotone since sum of monotone funcns.

$$\Rightarrow L_1(f, M) \leq \left\| f-g \right\|_1 \leq \int_0^1 L_1(f_{(t)}, M) \, dt$$
I. \( L_1(f, M) \leq \int_0^1 L_1(f_{(e)}, M) \, dt \):

\[
\forall t, \text{ let } g_{(e)} \leftarrow \text{ closest monotone fn to } f_{(e)}
\]

(Remember that \( f_{(e)} + g_{(e)} \) are 0/1 fns!)

\[
\text{let } g \leftarrow \int_0^1 g_{(e)} \, dt
\]

\( g \) is monotone since sum of monotone fns.

So

\[
L_1(f, M) \leq \| f - g \|_1
\]

\[
= \| \int_0^1 f_{(e)} \, dt - \int_0^1 g_{(e)} \, dt \|_1
\]

\[
= \| \int_0^1 f_{(e)} - g_{(e)} \, dt \|_1
\]

\[
\leq \int_0^1 \| f_{(e)} - g_{(e)} \|_1 \, dt = \int_0^1 L_1(f_{(e)}, M) \, dt
\]

II. \( L_1(f, M) = \int_0^1 L_1(f_{(e)}, M) \, dt \):

\[
\text{let } g \leftarrow \text{ closest mon fn to } f \text{ wrt } L_1
\]

so \( g_{(e)} \) mon \( \forall t \in [0, 1] \)
\[ L_1(f, M) = ||f - g||_1, \]
\[ = || \int_0^1 f(t) - g(t) \, dt ||_1, \]
\[ = \sum \int_0^1 \begin{cases} (f(t) - g(t)) & \text{if } f(t) > g(t) \\ 0 & \text{otherwise} \end{cases} \, dt + \sum \int_0^1 \begin{cases} (g(t) - f(t)) & \text{if } f(t) < g(t) \\ 0 & \text{otherwise} \end{cases} \, dt \]
\[ = \int_0^1 \begin{cases} \sum & \text{if } f(t) > g(t) \\ 0 & \text{if } f(t) = g(t) \end{cases} \, dt + \sum \int_0^1 \begin{cases} 0 & \text{if } f(t) > g(t) \\ (g(t) - f(t)) & \text{if } f(t) < g(t) \end{cases} \, dt \]

Claim: if \( f(t) \geq g(t) \) for all \( t \in [0, 1] \), then all terms in both sums are positive.

\[ = \int_0^1 ||f(t) - g(t)||_1 \, dt = \int_0^1 L_1(f(t), M) \, dt \]
Why is characterization helpful?

**Lemma**

If $T$ is a non-adaptive, $1$-sided error tester for $f : \mathbb{D} \rightarrow [0,1]$ w.r.t. Hamming measure, then always passes monotone $f$.

We have

- $f$ is monotone.
- $T$ queries points in $\mathbb{D}$ and passes iff $f$ monotone on $\mathbb{D}$.

Thus $T$ accepts $\Rightarrow$ $T$ accepts

(uses $1$-sided error of $T$)

if $f$ st. $L_1(f,M) = \epsilon n$:

**Lemma**

$L_1(f,M) = \int_0^1 L_1(f_{(t)},M) \, dt$

$\Rightarrow \exists \epsilon^* \text{ st. } L_1(f_{(\epsilon^*)},M) \geq \epsilon n$

Since $f_{(\epsilon^*)}$ is Boolean, Hamming dist. $+ L_1$ dist. of $f_{(\epsilon^*)}$ from monotone are the same!
\[ \Rightarrow T \text{ fails } f(x) \text{ for } \geq \frac{3}{4} \text{ choices of } Q : \]

i.e. with prob \( \geq \frac{3}{4} \), \( Q \) contains \( x \ll y \)

\[ \text{st.: } f(x) > f(y) \]

\[ \Downarrow \]

\[ f(x) \geq t^* > f(y) \]

\[ \Downarrow \]

So Q will output fail

\[ \text{maybe not a problem since algorithm only compares } f(x), f(y) ? \]

\[ \text{NO!!!} \]