Lecture 20:

Lower bounds via Communication Complexity

Linear functions:

↙ actually these are homomorphisms

f is "linear" iff $\forall x,y \quad f(x)+f(y)=f(x+y)$

will consider $f:\{0,1\}^d \to \{0,1\}$

here, "linear fncts" are the parity fctns (XOR)

<u>observation</u> $\forall x,y \quad f(x)+f(y)=f(x+y)$

iff

$$f(x)=\bigoplus_{i\in S} x_i \quad \text{for some } S\subseteq[d]$$

e.g. $\forall x, f(x)=0$ ← linear

$\forall x, f(x)=x\cdot b$ ← inner product

$\forall x, f(x)=1$ ⇐ not linear

K-linear fctns:

f is "k-linear" if

(1) linear

(2) depends on $=k$ variables ↙ also called "k-junta fctn"

i.e. $|S|=k$

linearity testing:

given $f: \{0,1\}^d \to \{0,1\}$ is f linear?

i.e. $\forall x,y \; f(x) + f(y) = f(x+y)$?

<u>Thm</u> Can propertytest linearity in $O(1)$ queries:

linearity test:

Pick random x, y + fail if $f(x) + f(y) \neq f(x+y)$

<u>Proof</u> later lecture

Consider functions $f: \{0,1\}^d \to \{0,1\}$ here, domain size $= 2^d \equiv n$

Testing k-linear functions: e.g. $f(x) = \bigoplus_{i \in S} x_i$ s.t. $|S| \le k$

related to testing if fctn is k-junta (depends only on k vars), low Fourier degree, computable by small depth decision trees, ...

<u>First Algorithm</u>: ("learns" f) wlog assume $f(\bar{0}) = 0$ else not linear

$O(d) = O(\log n)$

Query f on all $e_i = (00\ldots010\ldots0)$ for $i = 1..d$
$\uparrow$ i^{th} locn $\quad$ $\uparrow$ log n

$+ \ (00\ldots0)$

if $f(e_i) = 1$ for $\neq k$ i's then fail

else, test if

$f(x) = \underbrace{\bigoplus_{i \text{ s.t. } f(e_i) = 1}}_{\text{learned } f} x_i$ for most x via sampling

Can we do better?

What is Communication Complexity?

Shared randomness Setting:

$ Shared Randomness

Can talk over a channel

Alice has input $x = x_1 \cdots x_d$

Bob has input $y = y_1 \cdots y_d$

⋮

<u>Goal</u> Compute $f(x,y)$ ← how many bits, rounds of communication required?

<u>examples</u>:

1) $f(x,y) = \left(\bigoplus_i x_i\right) \oplus \left(\bigoplus_i y_i\right)$
   - requires 2 bits/round of communication
     $A \to B$ $\bigoplus_i x_i$
     $B \to A$ $f(x,y)$ (or $\bigoplus y_i$)

2) $f(x,y) = \sum x_i + \sum y_i$
   - requires $O(\log n)$ bits ← can we do better
     $A \to B$ $\sum x_i$
     $B \to A$ $\sum y_i$ (or $f(x,y)$)

3) $f(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{o.w.} \end{cases}$
   - requires $\Omega(\log d)$ bits with shared randomness

4) $f(x,y) =$ "do $x + y$ agree on any bit?"
   - requires $\Theta(d)$ bits

Communication Complexity (CC) lower bounds (we have these!)

$\Longrightarrow$ Property testing (PT) lower bounds

Idea give reduction from CC problem to PT problem

$\Longrightarrow$ L.B. for C.C. problem yields L.B. for P.T. problem

a lot of great work done in this area

so we get this almost for free!!

Example:

- A hard C.C problem

SET DISJOINTNESS

Alice	Bob
$X \in \{0,1\}^d$	$y \in \{0,1\}^d$

$$\mathrm{Disj}(x,y) = \bigvee_{i=1}^{d} (x_i \wedge y_i)$$

do A & B agree on any bit?

Known l.b.: $\Omega(d)$ bits of communication required to solve it.

even if allow many rounds, probabilistic protocols w/ shared randomness

Sparse Set disjointness: A & B have at most k 1's

needs $\Omega(k)$ bits communication
(even if guaranteed that intersect only once or not at all)

How can we use this to lower bound PT problems?

A reduction from sparse set disjointness to PT for 2k-linearity:

both Alice + Bob can query

$\boxed{\text{shared randomness}}$

Alice

Set A: n bit vector $\{0,1\}^d$ with exactly k 1's in it

describing k-linear fctn f

(i.e. f is XOR of bits with indices in A)

$\rightleftarrows$

Bob

n bit vector $\{0,1\}^d$ with k 1's: set B

describing k-linear fctn g

Question:

does $h = f \oplus g$

have $2k$-linearity property?

note: if $A \cap B = \phi$ then h is $2k$-linear

if $A \cap B \neq \phi$ then h is j-linear

for $j \leq 2k-2$.

e.g. if $A = \{X_1, X_2\}$ & $B = \{X_3, X_4\}$

$A \cap B = \phi$

$f = X_1 \oplus X_2$ $\quad g = X_3 \oplus X_4$

$h = X_1 \oplus X_2 \oplus X_3 \oplus X_4$ ← 4 linear

if $A = \{X_1, X_2\}$ $\quad B = \{X_2, X_3\}$

$A \cap B = \{X_2\}$

$f = X_1 \oplus X_2$ $\quad g = X_2 \oplus X_3$

$h = X_1 \oplus \underbrace{X_2 \oplus X_2}_{=1} \oplus X_3$

$= X_1 \oplus X_3$ ← 2 linear

for all X_i in $A \cap B$,

<u>two</u> variables drop out of h

so h is $(k - 2|A \cap B|)$-linear

Fact if $h_1 \neq h_2$ are 2 linear fctns (for any k)

then $\frac{\#x \text{ s.t. } h_1(x) \neq h_2(x)}{2^d} = \frac{1}{2}$

we will prove this soon

$\Rightarrow$ if $A \cap B = \emptyset$, h is $\frac{1}{2}$-far from $2k$-linear

Why is this interesting?

protocol for testing $2k$-linearity of h with q queries $\Rightarrow$ C.C. protocol for set disjointness of A, B

Shared random string which contains random bits for A's queries } R

A runs prop test alg. When needs $h(x) = f(x) \oplus g(x)$:

1) compute $f(x)$
2) ask Bob for $g(x)$
3) output $f(x) \oplus g(x)$ as $h(x)$

what is answer to my next question? + $f(x)$ $\longrightarrow$

$\longleftarrow$ $g(x)$

Bob simulates A's run on R.

Bob computes x & then $g(x)$

$\boxed{\text{Note: Alice doesn't need to sent } x\text{'s, just } f(x)\text{!!!}}$

d bits

1 bit

Total communication $= 2q$ bits

$\Rightarrow q = \Omega(k)$

Thm k-linearity testing requires $\Omega(k)$ queries!

Interesting, since linearity testing only needs $O(1)$!

<u>Proof of fact</u>: Given $h_1(x) = \bigoplus_{i \in S_1} x_i$ & $h_2(x) = \bigoplus_{i \in S_2} x_i$

if $h_1 \neq h_2$, $\exists\, i$ s.t. $i \in S_1 \Delta S_2$, wlog assume $i \in S_1$ & $i \notin S_2$

pair inputs $x, x' \in \{0,1\}^d$

s.t. $x = x' \oplus \underbrace{(0,0..0\overset{i\text{th bit}}{1}0\cdots0)}_{e_i}$

note $\forall$ pairs, $h_1(x) \neq h_1(x')$ since ith bit is different & $i \in S_1$

but $h_2(x) = h_2(x')$ since $i \notin S_2$

so <u>exactly one</u> of

$(h_1(x) = h_2(x))$ & $(h_1(x') = h_2(x'))$ hold

$$\Rightarrow \frac{\#\, x \text{ s.t. } h_1(x) = h_2(x)}{2^d} = \frac{1}{2} \quad \blacksquare$$