Linear functions:

\[ f \text{ is "linear" iff } \forall x, y \quad f(x+y) = f(x)f(y) \]

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

here, "linear" facts are the parity facts

observation \( \forall x, y \quad f(x)f(y) = f(x+y) \)

iff

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

K-linear functions:

\[ f \text{ is "K-linear" if (1) linear (2) depends on } k \text{ variables } \]

i.e. \( |S| = k \)

Also called "K-junta func"
Linearity testing:

Given \( f: \mathbb{R}^n \rightarrow \mathbb{R}^m \) is \( f \) linear?

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

E.g. \( \forall x \ f(x) = 0 \) is linear.
\( \forall x \ f(x) = 1 \) is not.
\( \forall x \ f(x) = x \cdot b \) inner prod of \( x \cdot b \) is linear.

Then can properly test linearity in \( O(1) \) queries:

Linearity test:
Pick random \( x, y \) + fail if \( f(x) + f(y) \neq f(x+y) \)

Proof later lecture
Consider functions $f : \{0,1\}^d \rightarrow \{0,1\}$ here, domain size $= 2^d = n$

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

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

First Algorithm: ("learns" $f$) wlog assume $f(0) = 0$

Given

Query $f$ on all $e_i = (00\ldots010\ldots0)$ for $i = 1, \ldots, d$

$\uparrow i^{th}$ locn

$\uparrow \log n$

$+ (00\ldots0)$

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

else, test if

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

via sampling

Can we do better?
What is Communication Complexity?

Settings:

Alice
has
input
\( x = x_1 \ldots x_n \)

Can talk
over a channel

Bob
has
input
\( y = y_1 \ldots y_n \)

Goal: Compute \( f(xy) \)

\( \text{how many bits, rounds of communication required?} \)

Examples:

1) \( f(xy) = (\oplus x_i) \oplus (\oplus y_i) \) • requires \( 2 \) bits/round of communication

\( A \rightarrow B \quad \oplus x_i \)
\( B \rightarrow A \quad f(xy) \) (or \( \oplus y_i \))

2) \( f(xy) = \sum x_i + \sum y_i \) • requires \( O(\log n) \) bits

\( A \rightarrow B \quad \sum x_i \)
\( B \rightarrow A \quad \sum y_i \) (or \( f(xy) \))

3) \( f(xy) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \) • requires \( O(\log n) \) bits

4) \( f(xy) = \text{"do } x \text{ and } y \text{ agree on any bit?"} \) • requires \( O(n) \) bits

\( \text{Can we do better?} \)
Communication Complexity (cc) lower bounds (we have these!)

⇒ Property testing (pt) lower bounds

Idea: give reduction from CC problem to PT problem

⇒ L.B. for CC problem yields L.B. for PT problem

so we get this almost for free!!

Example:

A hard CC problem

SET DISJOINTNESS

Alice

$X \in \{0,1\}^n$

Bob

$y \in \{0,1\}^n$

$\text{Disj}(x,y) = \bigvee_{x=1}^n (x_i \land y_i)$

do A+B agree on any bit?

Known L.B.: $\Omega(n)$ bits of communication required to solve it. even if allow many rounds, probabilistic protocols

Sparse Set disjointness: A+B have almost $k$ 1's

needs $\Omega(k)$ bits of 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 k-linearity:

both Alice and Bob can query

Shared randomness

Alice

Set A

\( n \) bit vector \( \{0,1\}^n \)
with exactly \( k \) 1s in it

describing \( k \)-linear fn \( f \)

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

Bob

n bit vector \( \{0,1\}^n \)
with \( k \) 1's describing \( k \)-linear fn \( g \)
Question:

does $h = f \oslash g$

have $2k$-linearity property?

Note:

if $A \cap B = \emptyset$ then $h$ is $2k$-linear.

if $A \cap B \neq \emptyset$ then $h$ is $j$-linear for $j \leq 2k-2$.

E.g., if $A = \{x_1, x_2^3\}$ and $B = \{x_3, x_4^3\}$

$A \cap B = \emptyset$

$f = x_1 \oslash x_2$

$g = x_3 \oslash x_4$

$h = x_1 \oslash x_2 \otimes x_3 \oslash x_4 \leftarrow 4$-linear

if $A = \{x_1, x_2^3\}$ and $B = \{x_1, x_3^3\}$

$A \cap B = \{x_2^3\}$

$f = x_1 \oslash x_2$

$g = x_1 \oslash x_3$

$h = x_1 \oslash x_2 \oslash x_3 \oslash x_3 \overset{=1}{=}$

$= x_1 \oslash x_3 \leftarrow 2$-linear

for all $x_i$ in $A \cap B$,

\two variables drop out of $h$

so $h$ is $(k-2|A \cap B|)$-linear.
Fact: if \( h_1 \neq h_2 \) are 2 linear functions (for any \( k \)),
then \[ \frac{\#x \text{ st. } h_1(x) \neq h_2(x)}{2^n} = \frac{1}{2} \]

We will prove this in homework.

\[ \Rightarrow \text{If } \{A,B\} = \emptyset, \text{ } h \text{ is } \frac{1}{2}-\text{far from } 2k-\text{linear} \]

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 both queries. Bob simulates A's run on R. Bob computes x & then

\( A \) runs prep test alg. When needs \( h(x) = f(x) @ g(x) \):
1) compute \( f(x) \)
2) ask Bob for \( g(x) \)
3) output \( f(x) @ g(x) \)
as \( h(x) \)

**Note:** Alice doesn't need to sent x's

Total communication = \( 2q \) bits

\[ \Rightarrow \quad q = O(k) \]

**Thm:** k-linearity testing requires \( O(k) \) queries!

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