Fundamental Techniques in Pseudorandomness
Part 1

Motivations and goals
(and previews of future lectures)
Goals:
- Introduce fundamental problems
- Introduce basic techniques
- Show how basic techniques can be composed to provide solutions

Message:
- everything follows from
  - elementary algebra
  - good definitions
  - composition

How elementary is the linear algebra:
- In every field:
  - \( k \) linearly independent linear equations in \( n \) variables have a solution space of dimension \( n-k \)
  - A non-zero degree-\( d \) polynomial in one variable has \( \leq d \) roots
What problems do we want to solve?

1. Construct efficiently and deterministically objects whose existence is guaranteed by proofs based on the probabilistic method.

2. Convert a randomized algorithm for a problem of interest into a deterministic algorithm of comparable complexity.

3. Efficiently construct deterministically (or with little randomness) objects having many of the useful properties of random objects.
Probabilistic Method: Example 1

Ramsey Theorem (Erdős, Szekeres)
Every n-vertex graph has either a clique or an independent set of size \( \frac{1}{2} \log n \)

Erdős
There is an n-vertex graph in which \( \max \text{ clique} \leq 2 \log n \) and \( \max \text{ i.s.} \leq 2 \log n \)

80-year old problem: match Erdős's existence proof with an explicit (say, polynomial time) construction

Recent breakthrough (Chattopadhyay, Zuckerman, Cohen):
\[
\max \text{ clique}, \max \text{ i.s.} \leq \exp \left( (\log \log n)^c \right)
\]
\( c \) absolute constant
Proportional Method: Example 2

Shannon's Second Theorem

\[
\begin{array}{c}
\text{A} \quad \text{n-bit} \quad \Rightarrow \quad \text{Noisy Channel} \quad \Rightarrow \quad \text{B}
\end{array}
\]

Pick \( C : \{0, 1\}^n \rightarrow \{0, 1\}^m \), known to both A and B.

A computes and sends \( C(x) \).

B receives corrupted transmission \( y \).

B guesses message as

\[
x' = \arg \min d_H (C(x), y)
\]

Thus (very informally):

Either this works or it is impossible for any coding scheme to reliably send an \( n \)-bits message by transmitting \( m \) bits over channel.

Note: \( C \) has doubly exponential size.

Brute force decoding takes exponential time.

Lots of work in past 70 years toward making code explicit and encoding and decoding polynomial (or even linear) time.
Example 3: One Time Pad

$c' = y \oplus k$

$c = x \oplus k$

$x = c \oplus k$

$A$

$B$

$G$
Desynchronization of Algorithms

Example 1: Primality testing

Idea of Miller-Rabin and Solovay-Strassen randomized algorithms:

If \( N \) is composite, it has ‘certificates of compositeness’ (e.g. violations of Fermat’s Little Theorem, or of Quadratic Reciprocity) that exist and can be found with high probability using randomness

To find a deterministic algorithm: construct certificates in polynomial time

AKS: new randomized algorithm.

- given \( N \), define polynomial \( p_N \) that is efficiently evaluable and non-zero iff \( N \) is composite
- find non-zero point of \( p_N \) randomly if one exists
- desynchronization: show how to construct a non-zero point of \( p_N \) if one exists
Derandomization of Algorithms

Example 2: Undirected Reachability

Problem: given undirected graph $G$, nodes $s, t$, is there a path from $s$ to $t$?

Aleliunas, Karp, Lipton, Lovász, Rackoff:

- start at $s$
- do a $100 \cdot n^3$-step random walk
- if $t$ is encountered, output YES
- else output NO

Memory use: $O(1)$ variables, $O(\log n)$ bits

Reingold: same performance, deterministically
Derandomization of Algorithms

Example 3: Finding Large Primes

Problem: given $N$, find a prime between $N$ and $2N$

Randomized algorithm: pick $O(\log N)$ random integers

Deterministic algorithm: ??
Decomposition of Algorithms

Example 4: Polynomial Identity Testing

Problem: given two multivariate polynomials $p, q$, e.g. as formulas or as arithmetic circuits, is $p = q$?

Randomized Algorithm: check a random point in a suitably large discrete range

Deterministic Algorithm: ??
  (cf. AKS)
Desynchronization of Algorithms

Example 5: Approximating the Permanent

Problem: given square 0/1 matrix $M$, approximate

$$\text{perm}(M) = \sum_{\pi} \prod_{i} M_{\pi(i), \pi(i)}$$

Equivalently: given bipartite graph, approximate the number of perfect matchings

(Approximate: achieve, say, 1% multiplicative approximation)

Randomized Algorithm: long story
(Jerrum, Sinclair, Vigoda)

Deterministic Algorithm: ??
Derandomization of Algorithms

Example 6: Circuit acceptance

Problem: given boolean circuit $C$ with one bit output, find a number in the range

$$\mathbb{P}_{x \sim U_n} \left[ C(x) = 1 \right] = \frac{1}{10}$$

Randomized algorithm: evaluate $C$ at 1,000 random inputs, output fraction of times you see 1

Deterministic algorithm: ??

Note: a derandomization of this algorithm implies a derandomization of all algorithms.
Complexity Theoretic Questions

$P = BPP$?

can we solve in polynomial time, deterministically, all problems that we can solve in polynomial time probabilistically whp?

Note: implied by derandomization of circuit approx problem

- also implied by plausible circuit complexity conjectures
- it implies unproven (and hard) circuit complexity conjectures

$L = BPL$?

Is every problem solvable in polynomial time and logarithmic space prob. whp also solvable in poly time and log space deterministically?

- Non-trivial result: can do in space $O((\log n)^{1.5})$, time $O(\log n)$
- No known barrier
Unifying Notion: Pseudorandom Generator

\[ G : \{0,1\}^t \to \{0,1\}^n \] \( \epsilon \)-fools a family of tests \( F \), where each \( f \in F \) is \( f : \{0,1\}^n \to \{0,1\} \)

\[
\begin{align*}
\text{RAND} & \xrightarrow{t \text{ bits}} G & \xrightarrow{n \text{ bits}} f^0 \\
\text{RAND} & \xrightarrow{n \text{ bits}} f^0 \\
\forall f \in F. \ & \left| \Pr_{x \sim U^n} [f(x) = 1] - \Pr_{z \sim U^t} [f(G(z)) = 1] \right| \leq \epsilon
\end{align*}
\]
The case of existence of PRGs:

Suppose we have polynomial-time computable

\[ g : \mathbb{N} \to \{0,1\}^n \]

Then, \[ P \subseteq \text{BPP} \]
- all applications of poly-time methods to construct objects with poly-time checkable properties can be made constructive
- all pseudorandomness algorithms can be formalized

\[ r = \begin{cases} \frac{1}{2} & \text{in } \frac{n}{k} \text{ steps, current } \text{ with } \text{error} \\ \frac{1}{n} & \text{in } \frac{n}{k} \text{ steps, current } \text{ with } \text{error} \end{cases} \]

Interested in constructing

\[ C \subseteq \text{BPP} \]

that satisfies property \( P \)

\[ P \] is checkable in \( n^2 \) time

\[ C \subseteq \text{BPP} \]

\[ x \in \mathbb{N} \]

\[ g(x) \]

\[ r \]

\[ \left\{ \begin{array}{l} 0^C \rightarrow 0 \text{ with probability } 1/2 \\ 0^C \rightarrow 1 \text{ with probability } 1/2 \end{array} \right. \]

\[ \frac{1}{2} \text{ of trials is } 0 \]

\[ z = \left\{ \begin{array}{l} 0^C \rightarrow 0 \text{ with probability } 1/2 \\ 0^C \rightarrow 1 \text{ with probability } 1/2 \end{array} \right. \]

\[ \text{prob that } g(x) \text{ is } 0 \text{ is } \frac{1}{2} \text{ of trials is } 0 \]
Part 2

Simple constructions of Pseudorandom Generators
Tests that look at only one bit construct:

\[ G : \{0,1\}^t \rightarrow \{0,1\}^N \]

such that, for uniform \( x \), each bit of \( G(x) \) is uniformly distributed
Tests that look at 2 bits

Construct

\[ G : \{0,1\}^t \rightarrow \{0,1\}^N \]

such that, for uniform \( x \), the bits of \( G(x) \) are uniformly distributed and pairwise independent

\[ N = 2^t - 1 \]

\( z \rightarrow \langle z, a_1 \rangle, \langle z, a_2 \rangle, \ldots, \langle z, a_{2^t - 1} \rangle \)

\( z \in \mathbb{F}_2^t \) where \( a_1, \ldots, a_t \)

is an enumeration of \( \mathbb{F}_2^t \)

\[ \langle z, a \rangle = 0 \land \langle z, a' \rangle = 0 \]

\( z_1, z_2, z_3 \rightarrow z_1, z_1 \oplus z_2, z_1 \oplus z_3, z_2 \oplus z_3, z_1 + z_2, z_1 + z_3, \]

\[ z_2 + z_3, z_1 + z_2 + z_3 \]
Tests that look at two values

Construct

\[ G : \{0,1\}^t \rightarrow \{0,1\}^n \rightarrow \{0,1\}^m \]

such that, for uniform \( x \), the outputs of \( h_x = G(x) \) are pairwise independent.

\[ n = m \]

\[ \mathbb{F}_2^n \ (\mathbb{Z}_2)^n \]

input of generator \( a,b \in \mathbb{F}_2^n \)

\[ t = 2n \]

\[ h_{a,b}(x) = ax + b \]

For every \( x \neq y \) over the rand. of \( a,b \)

\( ax + b \) are uniform

\( ay + b \) and indep.

what is prob. for fixed \( v,w \)

\( ax + b = v \) prob over \( a,b \)

\( ay + b = w \)

We can generate

\[ h : \{0,1\}^n \rightarrow \{0,1\}^m \rightarrow \{0,1\}^n \]

pairwise indep.

input is \( 2n \) bits

\[ m > n \]

\[ 2 : \max \{m, n\} \] bits
Tests that look at $k$ values construct

$G : \{0,1\}^k \to (\{0,1\}^n \to \{0,1\}^n)$

such that, for uniform $z$, $h_z : \mathbb{G}_t$ has $k$-wise independent outputs

\[ n = m \]
\[ k = 3 \]

$t = 3n$

input of $G \alpha, \beta, \gamma \in \mathbb{F}_2^n$

$h_{\alpha, \beta, \gamma}(x) = ax^2 + bx + c$

Fix $x, y, z$

Fix $u, v, w$

\[ \Pr \left[ \begin{array}{l}
    h_{\alpha, \beta, \gamma}(x) = u \\
    h_{\alpha, \beta, \gamma}(y) = v \\
    h_{\alpha, \beta, \gamma}(z) = w
\end{array} \right] = \left( \frac{1}{2^n} \right)^3 
\]

\[ \prod_{\alpha, \beta, \gamma} \left[ \begin{array}{l}
    ax^2 + bx + c = u \\
    \beta y^2 + by + c = v \\
    \gamma z^2 + \beta z + c = w
\end{array} \right] = \frac{1}{(2^n)^3} \]

In general

$\{0,1,4\}^n \to (\{0,1\}^n \to \{0,1\}^n)$

$k$-wise $\{0,1\}^n \to \{0,1\}^m$

$t = k \times \max \{n, m\}$
Suppose $f : \mathbb{R} \to \mathbb{R}$ is a polynomial of degree $n$.

**Theorem**

$E[f(x)] = E[f(g(x))]

So if $f$ is a polynomial, then $E[f(g(x))]$ is also a polynomial of degree $n$.

**Example**

Consider the function $f(x) = x^2$.

**Diagram**

![Diagram of a quadratic function](image)

A graph of $f(x) = x^2$ is depicted by a quadratic distribution.

**Proof**

Let $f(x) = x^2$ be a polynomial of degree $n$.

$E[f(x)] = \sum_{x=0}^{x=n} x^2 \cdot p(x)

where $p(x)$ is the probability mass function.

**Corollary**

If $f(x) = x^2$ is a quadratic function, then $E[f(g(x))]$ is also a quadratic function.

**Theorem**

Let $f, g: \mathbb{R}^n \to \mathbb{R}$ be functions.

- $\text{Var}(f(x)) = \text{Var}(g(x))$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$

**Proof**

Let $f, g: \mathbb{R}^n \to \mathbb{R}$ be functions.

- $\text{Var}(f(x)) = \text{Var}(g(x))$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$

**Theorem**

Let $f, g: \mathbb{R}^n \to \mathbb{R}$ be functions.

- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$

**Proof**

Let $f, g: \mathbb{R}^n \to \mathbb{R}$ be functions.

- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
- $E[f(x)g(x)] = E[f(x)]E[g(x)]$
Pseudorandomness for Bounded-depth Circuits

Suppose \( f : \{0,1\}^n \rightarrow \{0,1\} \)

is computed by a Boolean circuit with:
- AND, OR, NOT gates
- AND, OR gates have no bound on fan-in, fan-out
- depth \( \leq d \)
- size \( = s \)

Then [Braverman]

There are \( \ell, u : \{0,1\}^n \rightarrow \mathbb{R} \) of degree
\[ \leq (\log \frac{\delta}{\varepsilon})^{O_d(1)} \]
such that

\[ \forall x. \quad \ell(x) \leq f(x) \leq u(x) \]

\[ \exists \varepsilon \quad u(x) - \ell(x) \leq \varepsilon \]
Construcing Optimal Ramsey Graphs in Polynomial Time

Erdős: if \( c = 2 \log n \), then with positive probability a \( G_{n, \frac{1}{2}} \) graph has no clique of size \( c \) and no independent set of size \( c \).

Proof: \( E(\# \text{cliques of size } c) + E(\# \text{ind. set of size } c) < 1 \)

---

Fix \( \varepsilon : 10^{-t} \rightarrow 10^{-t}(\varepsilon) \) \( (\varepsilon) \)-wise independent

Enough to have \( t = \left( \frac{2 \log n}{e} \right) \log (\varepsilon) \)

\( = o(\log^3 n) \)

Interpret output of \text{Gen} as \( n \)-vertex graph, call resulting distribution \( G_{n, \frac{1}{2}} \)

\[ E(\# \text{clique of size } c) + E(\# \text{ind. set of size } c) < 1 \]

Find out: \( 2o(\log^3 n) \cdot n^{2 \log_2 n} = n^{O(\log n)} \)
Tests that take XORs

Construct

\[ G : \{0,1\}^k \rightarrow \{0,1\}^N \]

such that, for every \( a \in \{0,1\}^N \),

\[ \mathbb{P} \left[ \langle G(x), a \rangle = 1 \right] \leq \frac{1}{2} + \varepsilon \]

where operations are in \( \mathbb{F}_2 \)

Input \( a \in \mathbb{F}_2^k \), \( b \in (\mathbb{Z}_2)^L \)

ar's \( \rightarrow \langle a, b \rangle, \langle a^i, b \rangle, \ldots, \langle a^L, b \rangle \)

take

\[ \sum_{i \in S} \langle a^i, b \rangle \]

\[ = \langle \sum_{i \in S} a^i, b \rangle \]

\[ \mathbb{P} \left[ \langle \sum_{i \in S} a^i, b \rangle = 0 \right] \]

\[ = \mathbb{P} \left[ \sum_{i \in S} a^i = 0 \right] + \frac{1}{2} \left( 1 - \mathbb{P} \left[ \sum_{i \in S} a^i \right] \right) \]

\[ = \frac{1}{2} + \frac{1}{2} \mathbb{P} \left[ \sum_{i \in S} a^i = \frac{N}{2} \right] \]

\[ \leq \frac{1}{2} + \frac{1}{2} \frac{N}{2^k} \]

\[ \frac{1}{2} \cdot 2^k \rightarrow \frac{3}{2} \cdot 2^k \]

\[ \varepsilon = \text{Fool} \text{ \ Low \ test} \]

\[ \varepsilon = \frac{N}{2^k} \]

\[ \varepsilon = O \left( \log \frac{N}{2} \right) \]
Applications

Take $f : \mathbb{S}^n \to \mathbb{R}$

write as

$$f(x) = \sum \hat{f}(s) \cdot (-1)^{a \cdot x}$$

Suppose $G : \mathbb{S}^n \to \mathbb{S}^n$ is $\varepsilon$-biased

Then

$$\forall a, \quad \left| \mathbb{E}_{U_n} (-1)^{a \cdot x} - \mathbb{E}_{U_E} (-1)^{a \cdot G(z)} \right| \leq 2 \varepsilon$$

So

$$\left| \mathbb{E}_{U_n} f(x) - \mathbb{E}_{U_E} f(G(z)) \right| \leq 2 \varepsilon \sum \hat{f}(s)$$

K-wise independence: fools $f$ of degree $k$ (no error)

$\varepsilon$-biased: fools $f$ of small $L_1$ error $\varepsilon \cdot \|f\|_L$

Cauchy-Schwarz

If $f : \mathbb{S}^n \to \mathbb{S}^n$ depends on $k$ variables

$$\sum \hat{f}(s) \leq \sqrt{2^k}, \quad \sqrt{\sum \hat{f}^2(s)} \leq \sqrt{2^k}$$

Often, if $f$ is fooled by $k$-wise independence, it is also fooled by $\varepsilon$-biased generators with $\varepsilon = \sqrt{2^k}$

Better: seed goes from $k \log n$ to $2^k + \log n$
\[ f = \text{depth } k \text{ decision tree} \]
\[ \text{k-wise ind. } \mathcal{F} \text{ fool } f \]
\[ t = k \cdot \log n \]

\[ f = \text{decision tree of size } \leq S \]
\[ \| f \|_1 \leq S \]
\[ f = \text{depth } k \text{ d.t.} \]
\[ \text{size } \leq 2^k \]
\[ \varepsilon/2^k \text{- bias generator, } \varepsilon \text{- fool } f \]
\[ t = \log \frac{n \cdot 2^k}{\varepsilon} = k + \log \frac{n}{\varepsilon} \]

\[ f \text{ comp. d.t. of depth, } O(\log n) \]
\[ \text{size poly}(n) \]

\[ O(\log n) \text{- wise indep. } \]
\[ 1/\text{poly}(n) \text{- bias } \]
\[ t = O(\log^2 n) \]
\[ t = O(\log n) \]
Optimal Ramsey graphs in time $n^{100\log n}$

Take $Gen : 30.14 \to 30.14^{(2)}$

to be $\epsilon$-biased with $\epsilon = \exp(100 \log n)$

Interpret output of $Gen$ as graph,
call $\tilde{G}_{n,\frac{1}{2}}$ resulting distribution

$$
\mathbb{E}_{G \sim \tilde{G}_{n,\frac{1}{2}}} \#(\log n) \text{-cliques in } G
$$

$$
= \sum_{S \subseteq V : |S| \leq \log n} \mathbb{E} I_{S \text{ is a clique in } G}
$$

$$
\leq \sum_{S \subseteq V : |S| \leq \log n} \left( \mathbb{E} I_{S \text{ is a clique in } \tilde{G}_{n,\frac{1}{2}}} + \epsilon \cdot 2^{\log n} \right)
$$

$$
\leq \frac{Erdos}{\tilde{G}_{n,\frac{1}{2}}} + \epsilon \cdot n \cdot \log n \cdot n \cdot \log n
$$

$$
\leq Erdos + \epsilon \cdot n \cdot \log n.
$$

---

$f : 30.14^n \to 30.14$

$f \circ \epsilon C_5 \times 5$

$\epsilon$-bias distrib

$\epsilon \cdot \sum_{S \subseteq 5} |c_S| = \operatorname{cool} f$
Error-correcting codes

\[ C : \Sigma^k \rightarrow \Sigma^n \]

is an error-correcting code of minimum distance \( \geq \Delta \) if

\[ \forall x \neq y \in \Sigma^k \]

\[ d_H (C(x), C(y)) \geq \Delta \]

Motivating application

\[ \begin{array}{c}
A \\
\xrightarrow{x \in \Sigma^k} \\
\downarrow C(x) \\
\text{Channel} \\
\xrightarrow{z} B \\
\end{array} \]

Find \( x' \) such that

\[ d_H (C(x'), 2) < \Delta / 2 \]

Channel: can be used to transmit \( n \) elements of \( \Sigma \)

- is guaranteed to make \( < \Delta / 2 \) errors
Linear Error-Correcting Codes

\[ C : \mathbb{F}^k \rightarrow \mathbb{F}^n \]
- is linear
- is an error-correcting code
  of minimum distance \( \geq \Delta \)

Possible to use linear algebra to
reason about encoding, decoding

E.g.
cell \( 1 \cdot y \): \# non-zero entries of \( y \)

Then
1. \( d_H (y, z) = \left| y - z \right| \)
2. \( C \) has min distance \( \geq \Delta \)
   iff
   \( \forall x \in \mathbb{F}^k, \left| C(x) \right| \geq \Delta \)

Proof
\[
\min_{x \neq y} \left| C(x) - C(y) \right| = \min_{x \neq y} \left| C(x - y) \right|
\]
\[
= \min_{x \neq 0} \left| C(x) \right|
\]

3. There is matrix \( M \) such that
   \( C(x) = M \cdot x \)
   Also, there is matrix \( P \) such that
   \( y \in \{ C(x) : x \in \mathbb{F}^k \} \iff P \cdot y = 0 \)
   (other \( P \) or \( M \) determines \( C \))

Note: \( C \) has min-distance \( \geq \Delta \)
   iff
   every \( \Delta + 1 \) rows of \( P \) are linearly independent
Reed-Solomon Codes

\[ C : F^k \rightarrow F^n \quad \left[ n \leq |F| \right] \]

choose \( a_1, \ldots, a_n \in F \)

given \( x_0, \ldots, x_{k-1} \) let \( p_x(z) = z^{k-1} x_{k-1} + \ldots + z x_1 + x_0 \)

\[ C(x) = p_x(a_1), p_x(a_2), \ldots, p_x(a_n) \]

\[ \text{min distance} \geq n - k + 1 \]
Hadamard code

\[ C : \mathbb{F}_2^t \rightarrow \mathbb{F}_2^{2^t} \]

Let \( a_1, \ldots, a_{2^t} \) be the elements of \( \mathbb{F}_2^t \)

\[ C(x) = \langle x, a_1 \rangle, \ldots, \langle x, a_{2^t} \rangle \]

\[ \min \text{ distance} = \frac{1}{2} \cdot 2^t \]

\[ C(x) - C(y) = \langle x - y, a_1 \rangle, \ldots, \langle x - y, a_{2^t} \rangle \]
Concatenation

Suppose we have

\[ C_1 : \Sigma^k \to \Sigma^N \]
\[ \text{min distance } \Delta_1 \]

\[ C_2 : \Sigma \to 3014^n \]
\[ \text{min distance } \Delta_2 \]

Then

\[ C : \Sigma^k \to 3014^n N \]

has min distance \( > \Delta_1 + \Delta_2 \)

---

E5: Concatenate Reed-Solomon

\[ RS : F_2^k \to F_2^n \]

with \( H : F_2^k \to F_2^{k\epsilon} \)

we get

\[ C : 3014^n F_2^k \to 3014^n N \]

with min distance \( > \frac{1}{2} \cdot (n-k) \cdot 2^\epsilon \)

choose \( n = \frac{k}{\epsilon} \), \( 2^k \geq n \)

\[ C : 3014^n \times 2^k \to 3014^n 2^k \times 2^k \]

\[ C : 3014^n \to 3014^n \quad n \geq \frac{k^2}{\epsilon^2} \]

min distance \( > n \cdot (\frac{1}{2} - \epsilon) \)

max distance \( \leq \frac{n}{2} \)
Linear Error-Correcting Codes

vs k-wise independence

Suppose $C : \mathbb{F}^t \rightarrow \mathbb{F}^n$

is linear e.c.c. of min dist $\geq k+1$

Let $P$ be $(n-k) \times n$ matrix s.t.

$Py = 0$ iff $y \in \text{Im}(C)$

If $\|y\| \leq k$, then $Py \neq 0$

Every $k$ columns of $P$ are linearly ind.

Consider $G : \mathbb{F}^{n-t} \rightarrow \mathbb{F}^n$ given by

$x \rightarrow P^T x$

$x \rightarrow P^T x, P^T x, ..., P^T x$

This is $k$-wise indep. because

every $k$ output bits correspond
to multiplying $x$ by linearly indep. vectors

Code of min distance 2

$x_1, ..., x_{n-1} \rightarrow x_1, ..., x_{n-1}, \sum_{i=1}^{n-1} x_i$

$C : \mathbb{F}^{n-1} \rightarrow \mathbb{F}^{n-1}$

$P = (1, ..., 1)$

$b \rightarrow b, b, ..., b$
Linear Error-Correcting Codes

vs $\varepsilon$-biased spaces

Suppose $C : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n$

is such that $\forall y \in \text{Im}(C), \ y \neq 0$

$$(\frac{1}{2} - \varepsilon) \cdot n \leq |y| \leq (\frac{1}{2} + \varepsilon) \cdot n$$

if $C(x) = \Pi \cdot x$

$$M_i = \left( \begin{array}{c} -M_{i-1} \\ -M_{n+1} \\ \vdots \\ -M_n \end{array} \right)$$

picking at random a row of $M$ gives an $\varepsilon$-biased space generator

$1011, 1100, \ldots, 1101 \rightarrow 30114K$

Fix $a \in 30114K$

pick at random $i \in [k], \rightarrow n, \varepsilon$-close to unbiased

consider $(M_i, a)$

pick a random $i$

consider $i$-th bit of $Ma$

$\varepsilon$-close to unbiased

consider vector $Ma = C(a)$

$n$-dim vector

$\geq (\frac{1}{2} - \varepsilon) \cdot n$ ones

$\leq (\frac{1}{2} + \varepsilon) \cdot n$ ones
Samples

Given oracle access to $f: \{0,1\}^n \to [0,1]$

Output an estimate $A$ of $E_{x \sim U_n} f(x)$

such that

$$\Pr \left[ \left| A - E_{x \sim U_n} f(x) \right| > \varepsilon \right] \leq \delta$$

randomness of algorithm

How many queries and how much randomness do we need as a function of $n, \varepsilon, \delta$?
Solution 1: independent queries

Make \( t \) independent queries \( x_1, \ldots, x_t \),
output \( \hat{A} := \frac{1}{t} \sum_i f(x_i) \)

Chernoff bound:

\[
P \left[ \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x) \right| > \varepsilon \right] \leq 2e^{-\frac{\varepsilon^2 t}{2}}
\]

choose \( t = \Theta \left( \frac{1}{\varepsilon^2 \log \frac{1}{\delta}} \right) \)

Complexity [ignore constants]

Queries: \( \frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \)

Randomness: \( n \cdot \frac{1}{\varepsilon^2} \log \frac{1}{\delta} \)
Solution 2: Pairwise Independent Queries

Generate $t$ pairwise independent elements $x_1, \ldots, x_t$ of $\text{Unif}(0, 1)^n$

Output $A := \frac{1}{t} \sum_i x_i$

Chebyshev Inequality

$$\Pr \left[ \left| A - \mathbb{E} f(x) \right| > \varepsilon \right] \leq \frac{1}{\varepsilon^2 t}$$

Take $\varepsilon = \frac{1}{\varepsilon^2 \delta}$

Complexity

<table>
<thead>
<tr>
<th>Method</th>
<th>Queries</th>
<th>Randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>$\frac{1}{\varepsilon^2} \log \frac{1}{\delta}$</td>
<td>$n \cdot \frac{1}{\varepsilon^2} \log \frac{1}{\delta}$</td>
</tr>
<tr>
<td>Pairwise ind.</td>
<td>$\frac{1}{\varepsilon^2} \cdot \frac{1}{\delta}$</td>
<td>$n + \log \frac{1}{\varepsilon} + \log \frac{1}{\delta}$</td>
</tr>
</tbody>
</table>
New tool: Random Walks on Expanders

Let $H = (V, E)$ be a $d$-regular graph.

$d = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ be eigenvalues of the adjacency matrix.

$\lambda = \max \{ |\lambda_2|, |\lambda_3|, \ldots, |\lambda_n| \}$

[we can get $\lambda = \Theta(\sqrt{d})$]

Let $f: V \to [0,1]$ be arbitrary.

Let $x_1, x_2, \ldots, x_t$ be the sequence of vertices encountered by taking a $(t-1)$-step random walk in $H$ ($x_1$ is uniform).

[log(|V|) + $(t-1) \log d$ random bits used]

THEN (Chernoff bound on expanders)

$$\Pr \left[ \left| \frac{1}{t} \sum_{i=1}^{t} f(x_i) - \mathbb{E}[f(x)] \right| > \varepsilon + \frac{1}{d} \right] \leq e^{-\Omega(\varepsilon^2 t)}$$

(random walk)
Solution 3: Random Walk on Expanders

Construct \( H = (\{0, 1\}^n, E) \) such that
\[
\chi_d \leq \frac{\varepsilon}{\delta}
\]
(Enough to take \( d = \Theta(\varepsilon^2) \))

Pick a random walk \( x_1 \ldots x_t \) in \( H \)

Output \( A := \frac{1}{t} \sum_{i=1}^{t} f(x_i) \)

\[
P \left[ |A - Ef| > \varepsilon \right]
= P \left[ |A - Ef| > \frac{\varepsilon}{\delta} + \frac{\chi_d}{\delta} \right] \leq e^{-\Omega(\varepsilon^2 t)}
\]

Enough to take \( \varepsilon = O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}) \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Queries</th>
<th>Randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>( \frac{1}{\varepsilon^2} \log \frac{1}{\delta} )</td>
<td>( n \cdot \frac{1}{\varepsilon^2} \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td>Pairwise ind.</td>
<td>( \frac{1}{\varepsilon^2} \cdot \frac{1}{\delta} )</td>
<td>( n + \log \frac{1}{\varepsilon} + \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td>Expanders r.w.</td>
<td>( \frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} )</td>
<td>( n + \frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log \frac{1}{\varepsilon} )</td>
</tr>
</tbody>
</table>
Solution 4: composition