# Some Open Problems in Coding Theory

# Plan

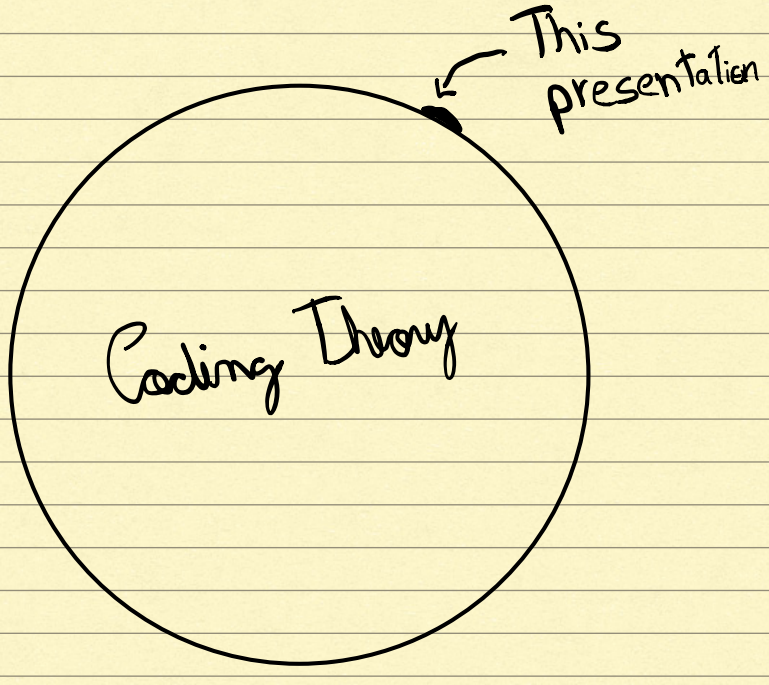
- Brief overview of some aspects of Coding Theory

(10 minute break)

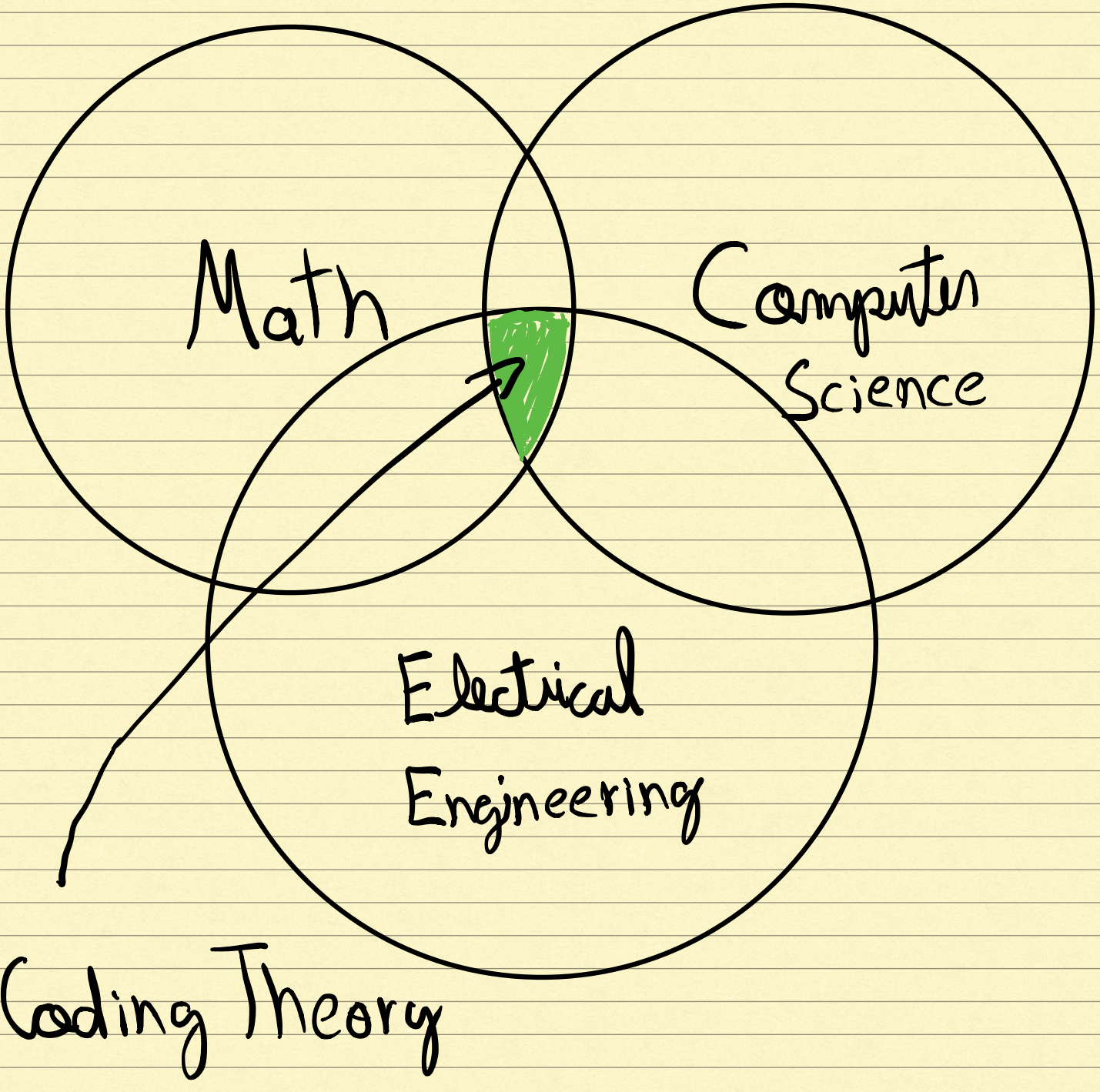
- Open Problems

# Disclaimer

- The choice of topics is biased towards my interests and knowledge of the field.
- The problems are open to the best of my knowledge.
- Some open problems are well-known longstanding problems (possibly challenging).



# Coding Theory at a Macroscopic Scale



# Intersections

Combinatorics

Expanders/HDXs

SOS


Local Algorithms
LDPCs

Hardness
PCPs

Algebraic
Methods

Highly
Parameter
Oriented

Probabilistic
Methods

# What is a code?

$\Sigma$ alphabet with $q$ symbols

(e.g. $\Sigma = \{0, 1, \ldots, q-1\}$)

$n \in \mathbb{N}$ the block length

A code is a subset

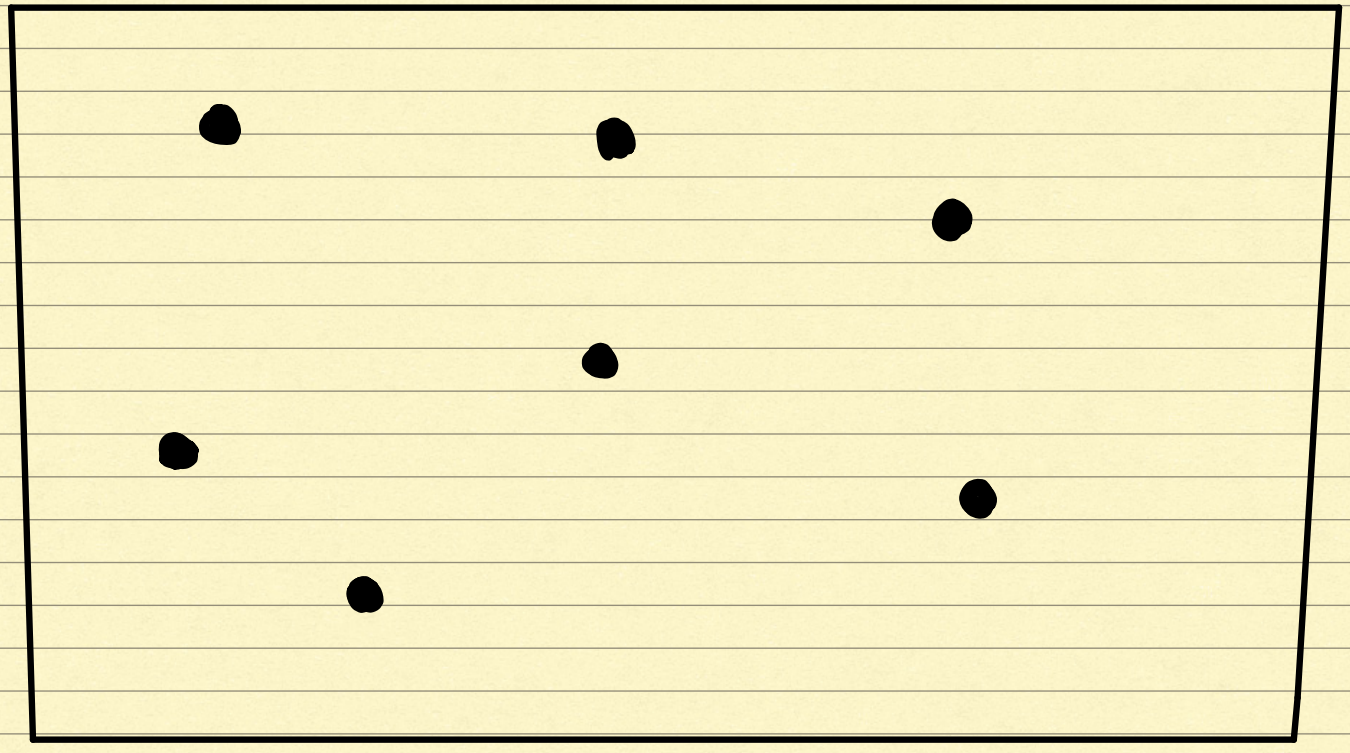
$$\mathcal{C} \subseteq \Sigma^n.$$

Coding Theory is essentially the study of properties of (sets of) strings.

aka codewords

# Pictorial View

$\mathcal{C} \subseteq \Sigma^n$



- codewords of $\mathcal{C}$

(a bit too unstructured)

# Adding a Metric

- Hamming Distance

$$x, y \in \Sigma^n$$

$$\Delta(x, y) := \frac{\sum_{i=1}^{n} \mathbb{1}_{[x_i \neq y_i]}}{n}$$

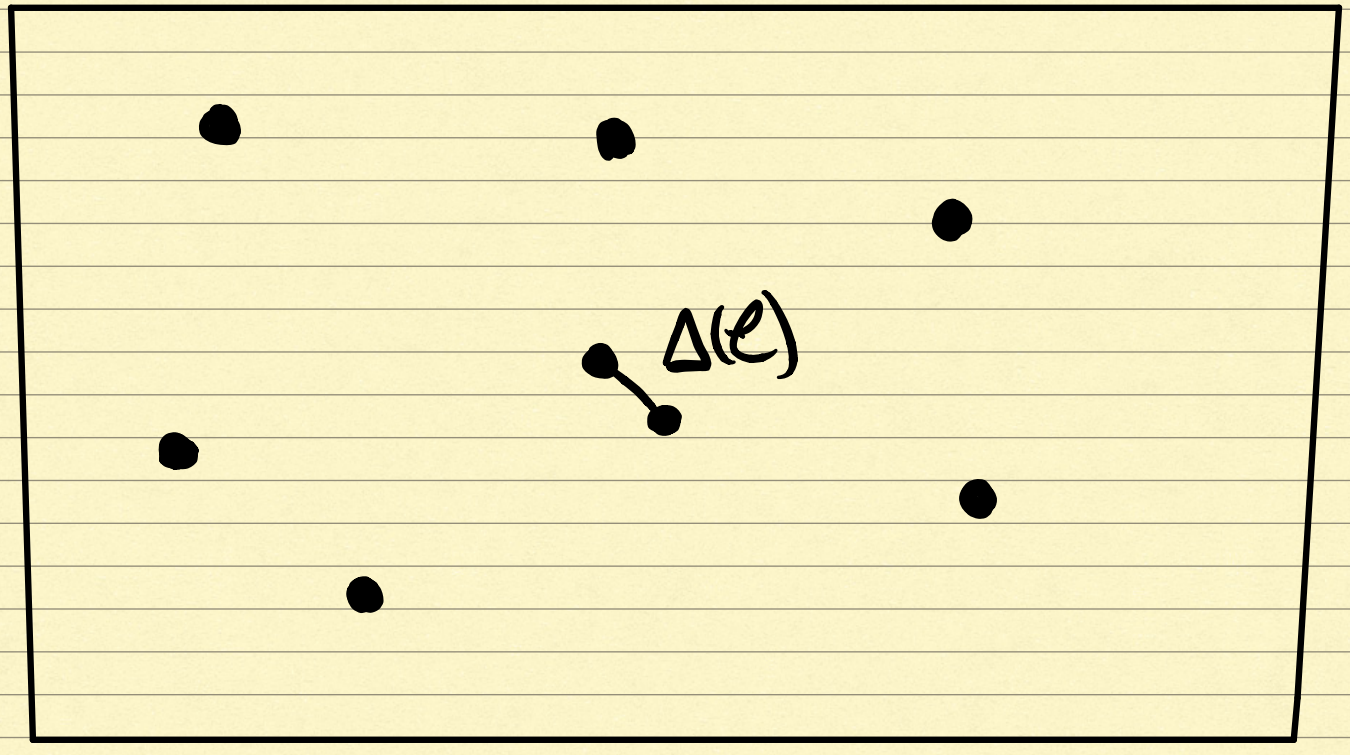
Ex: $x = 0\ 1\ 0\ 1\ 1\ 1$

$y = 1\ 1\ 1\ 1\ 1\ 1$

$$\Delta(x, y) = 2/6$$

# Minimum Distance

$\mathcal{C} \subseteq \Sigma^n$




- codewords of $\mathcal{C}$

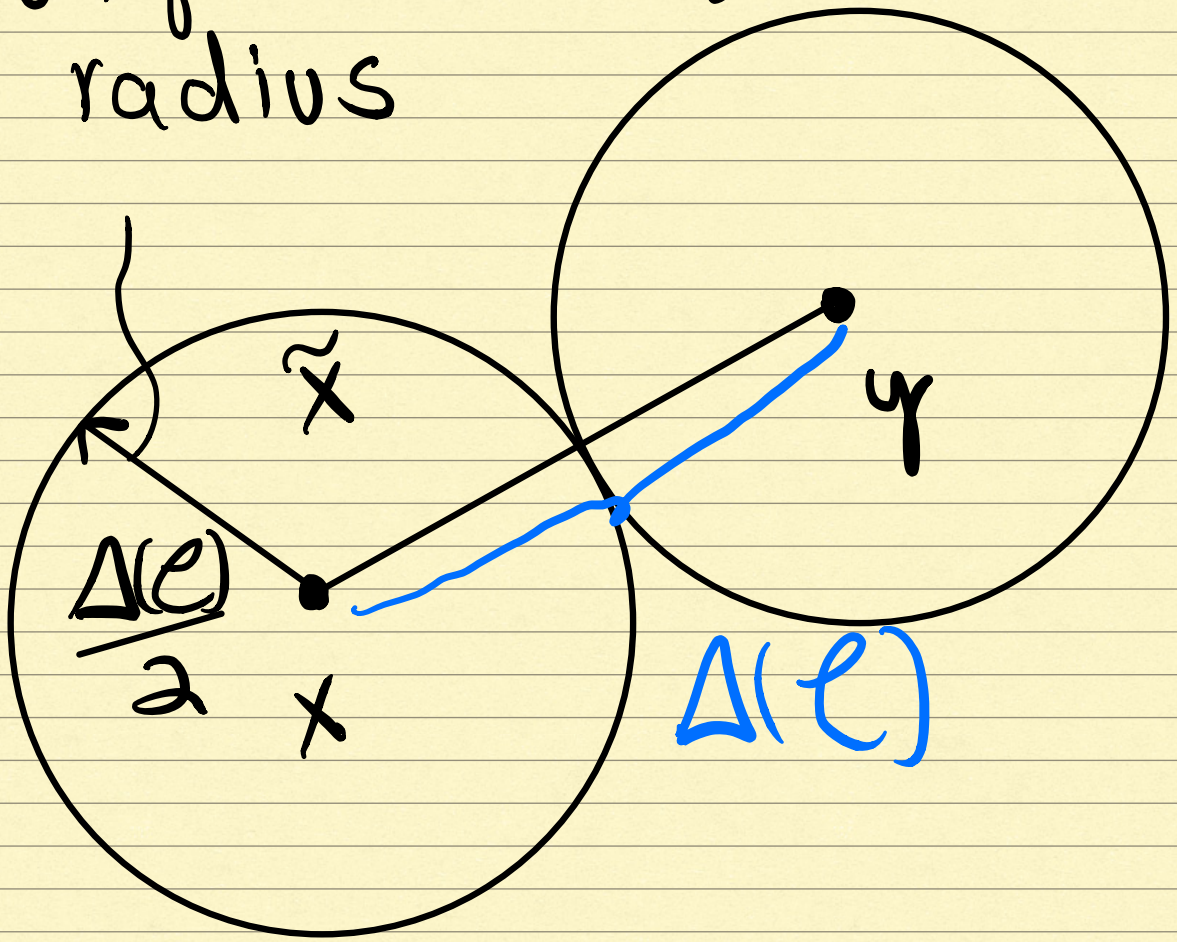
The (min.) distance of a code $\mathcal{C}$

$$\Delta(\mathcal{C}) := \min_{\substack{x \neq y \\ x, y \in \mathcal{C}}} \Delta(x, y)$$

# Why is $\Delta(\mathcal{C})$ important?

$x, y \in \mathcal{C}$ with $\Delta(x,y) = \Delta(\mathcal{C})$

Unique decoding radius




$x$ corrupted to $\tilde{x} \in B(x, \Delta(\mathcal{C})/2)$

# Achieving Large $\Delta(\mathcal{C})$

Ex: $\Sigma = \{0,1\}$

$$\mathcal{C} = \{\underbrace{00\ldots0}_{n \text{ times}}, \underbrace{11\ldots1}_{n \text{ times}}\}$$

$$\sum_{n}^{\infty}$$

$\Delta(\mathcal{C}) = 1$ ✓ (great)

$|\mathcal{C}| = 2$ ∧ (bad)

# Rate of $\mathcal{C}$

Want $|\mathcal{C}|$ as large as possible.

The rate of $\mathcal{C}$ is

$$r(\mathcal{C}) := \frac{\log_{|\Sigma|}(|\mathcal{C}|)}{n}.$$

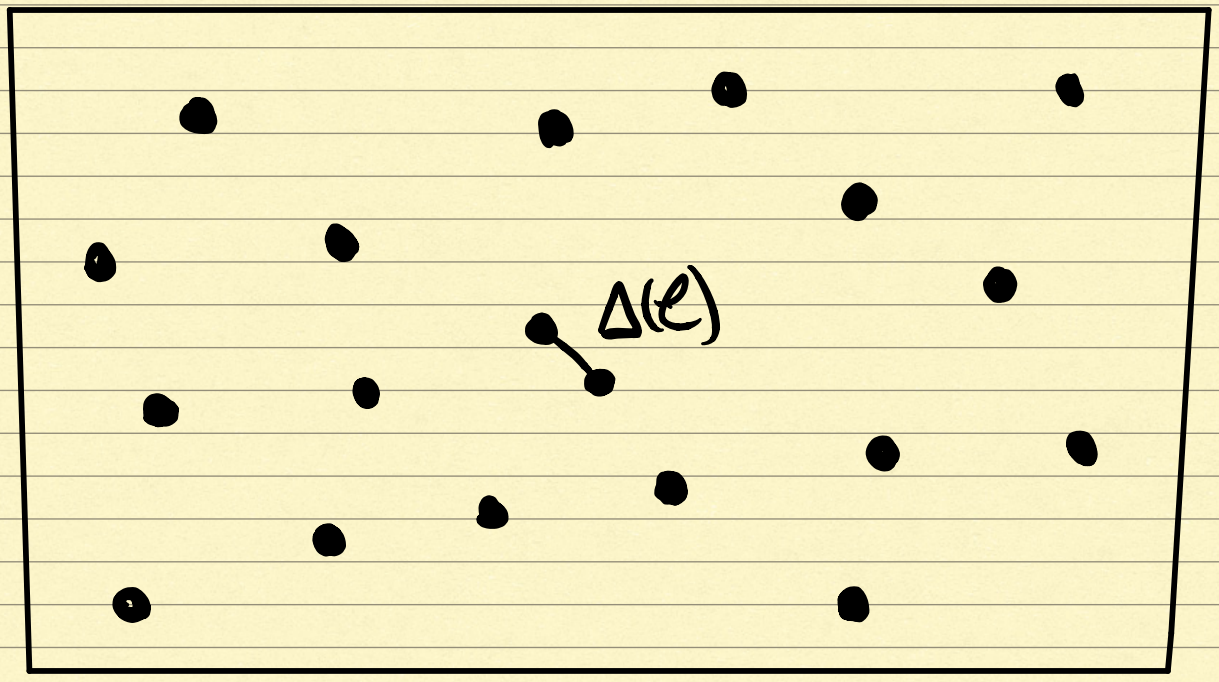
(the larger the better)

# Conflicting Goals

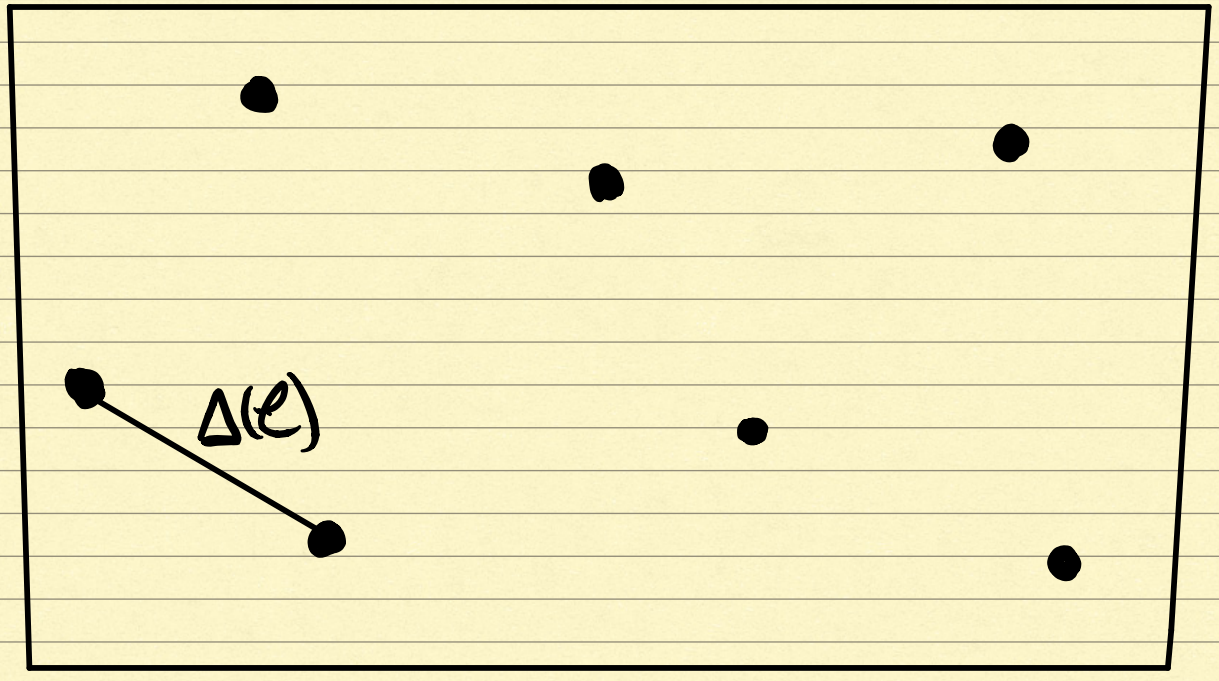
For a code $\mathcal{C}$

- Want $\Delta(\mathcal{C})$ large, and
- Want $r(\mathcal{C})$ large.

High $r(\ell)$ causes small $\Delta(\ell)$




High $\Delta(\ell)$ causes small $r(\ell)$



# Fundamental Question

"What is the best trade-off between rate and distance of a code?"

Much of coding theory research (for the Hamming model) revolves around this question.

# Wish List for a Code

Ideally, we may want a subset of:

- $\mathcal{C}$ is explicit
- $\mathcal{C}$ achieves the best parameter trade-off
- $\mathcal{C}$ is efficiently encodable
- $\mathcal{C}$ is // decodable/ list decodable
- $\mathcal{C}$ has some local properties
- $\mathcal{C}$ is over a small alphabet

# Examples of Codes

To illustrate we will talk about:

— Reed-Solomon Codes

— Hadamard Codes

— Expanders and Codes

# Some Notation

$$\mathcal{C} \subseteq \Sigma^n$$

For convenience, suppose $|\mathcal{C}| = |\Sigma|^k$ for some integer $k$.

Let $\mathcal{M} = \Sigma^k$ be the message space.

We may want an encoding function $Enc : \mathcal{M} \to \mathcal{C}$ (Enc is simply a bijection).

# Linear Codes

(possibly the most important class of codes)

$\Sigma = \mathbb{F}$ finite field

**Def. (Linear Code)**

$\mathcal{C} \subseteq \mathbb{F}^n$ is a linear subspace.

$$n \cdot r(\mathcal{C}) = \dim(\mathcal{C}) =: k$$

$\mathcal{C}$ can be specified either by

- a "generator matrix" $G \in \mathbb{F}^{n \times k}$
  s.t. $\mathcal{C} = \{Gz \mid z \in \mathbb{F}^k\}$
- a "parity check matrix" $H \in \mathbb{F}^{k \times n}$
  s.t. $\mathcal{C} = \{x \in \mathbb{F}^n \mid Hx = 0\}$

**Def** (Hamming weight)
Let $\Sigma = \{0, 1, \ldots, q-1\}$ and $x \in \Sigma^n$.
The Hamming weight $\|x\|$ of $x$ is

$$\|x\| = \sum_j \mathbb{1}[x_j \neq 0]/n.$$

Claim: If $\mathcal{C}$ is linear, then

$$\Delta(\mathcal{C}) = \min_{0 \neq x \in \mathcal{C}} \|x\|.$$

Proof Follows from

$$\Delta(x, y) = \Delta(x-y, 0) = \|x-y\|,$$

but $x - y \in \mathcal{C}$ since $\mathcal{C}$ is linear.

□

# Reed-Solomon Codes

Parameters

- $n$ the block length
- $k$ message length (rate $k/n$)
- $\mathbb{F}$ a field of size $\geq n$
- $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}$ distinct

Set of degree $k-1$ polynomials

$$\mathcal{P}_{k-1} := \{ p(x) = c_0 + c_1 x + \ldots + c_{k-1} x^{k-1} \mid (c_0, c_1, \ldots, c_{k-1}) \in \mathbb{F}^k \}$$

Reed-Solomon Code

$$\mathcal{C} = \{ (p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_n)) \mid p \in \mathcal{P}_{k-1} \}$$

# Well-known mantra

**Fact:** $(\forall 0 \neq p \in \mathcal{P}_{k-1})$ $(p$ has $\leq k-1$ roots$)$.

This fact implies amazingly large distance for Reed-Solomon.

**Proof**

$\forall p, q \in \mathcal{P}_{k-1}$ with $p \neq q$

$0 \neq p - q \in \mathcal{P}_{k-1}$

abuse of notation

$n\,\Delta(p,q) = n\|p-q\| \geq n - \#\{\text{roots of } p-q\}$

$\Downarrow$

$n\,\Delta(\mathcal{C}) \geq n - k + 1$

$\blacksquare$

# Parameters of Reed-Solomon

$$\left[ n, \frac{k}{n}, \frac{n-k+1}{n} \right]_{|F|}$$

block length

rate

distance

alphabet size

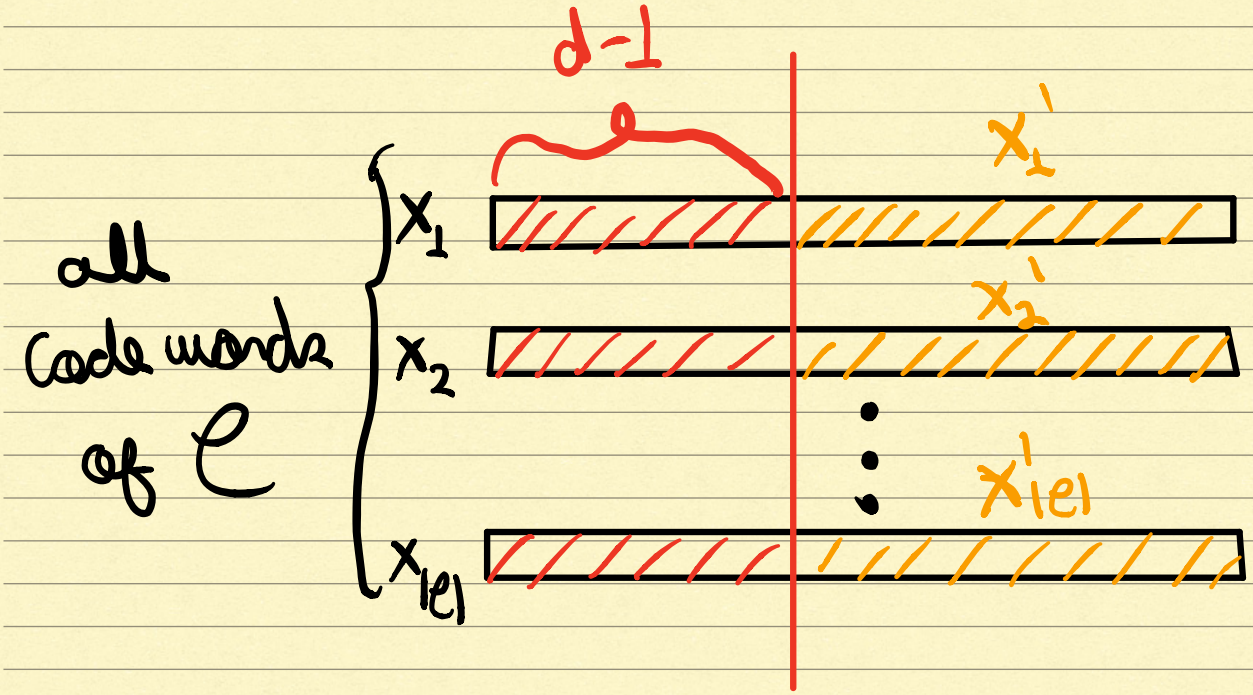

$|F| \geq n$

large alphabet

We will see that this trade-off is optimum

# Singleton Bound

Let $\mathcal{C}$ be $[n, \frac{k}{n}, \frac{d}{n}]_q$




delete the first $d-1$ symbols

$x_1', \dots, x_{|\mathcal{C}|}'$ are distinct since $\Delta(\mathcal{C}) \geq d$

$$\Downarrow$$

$$q^k = |\mathcal{C}| \leq q^{n-(d-1)} \Longrightarrow \boxed{k \leq n-d+1}$$

# Hadamard Codes

For $x \in \mathbb{F}_2^k$, define the linear function

$$f_x(y) = \langle x, y \rangle$$
$$= \sum_i x_i y_i \pmod 2.$$

Define the evaluation table of $f_x$ as

$$\mathrm{Eval}(f_x) = (f_x(y))_{y \in \mathbb{F}_2^k}$$

The Hadamard code is

$$\mathrm{Had}_k := \{ \mathrm{Eval}(f_x) \mid x \in \mathbb{F}_2^k \}.$$

Its block length is $n = 2^k$ and

$$|\mathrm{Had}_k| = 2^k.$$

The rate $r(Had_k) = \frac{K}{2^k}$

$= \frac{\log n}{n}$.

To analyze $\Delta(Had_k)$ we will use a bit of Fourier analysis.

$f_x, f_y$ with $x \neq y$

$f_x + f_y = f_{x+y}$ (Property of linear function)

$\Delta(f_x, f_y) = \| f_{x+y} \|$ (since $Had_k$ is a linear code)

$$\sum_{z \in \mathbb{F}_2^k} \frac{(-1)^{\langle x+y, z \rangle}}{n} = 0 \quad \text{for } x \neq y$$

$$\Rightarrow \Delta(f_x, f_y) = \frac{1}{2} .$$

Had$_k$ parameters $(k = \log_2 n)$

$$\left[ n, \frac{\log_2 n}{n}, \frac{1}{2} \right]_2$$

block length — rate — distance — alphabet size

rate: (exp. small)

distance: ✓ (large)

alphabet size: ✓ (small)

# Local Properties

Hadamard codes have amazing local properties.

"informal"

**Def.** (Locally Testable Codes)

We say that a code $\mathcal{C} \subseteq \Sigma^n$ is an $\ell$-LTC if $\exists$ a tester $\tau$ s.t. $\forall x \in \Sigma^n$, $\tau$ queries $\leq \ell$ positions of $x$ and

- (Completeness) if $x \in \mathcal{C}$, then $\tau$ accepts w.p. $\geq 2/3$;
- (Soundness) if $\Delta(x, \mathcal{C}) = \Omega(1)$, then $\tau$ accepts w.p. $\leq 1/3$.

Consider the following tester $\tau$ for $Had_k$:

Input: oracle to $f: \mathbb{F}_2^k \to \mathbb{F}_2$

$\tau$ samples $y_1, y_2 \in_U \mathbb{F}_2^k$,

queries $f(y_1), f(y_2), f(y_1+y_2)$

accepts iff $f(y_1)+f(y_2)=f(y_1+y_2)$.

**Theorem (BLR linearity test)**

$Had_k$ is a 3-LTC with tester $\tau$ (above).

Proof Exercise. Hint: Fourier analysis

# Expanders and Codes

Roughly, expanders have two main uses in Coding Theory:

- Distance amplification

- Define the parity check matrix (Tanner codes)

# What is an expander?

Expander: "well connected sparse graph"

"derandomization of a complete graph"

"random like explicit sparse graph"

Roughly, we will have:

— Being sparse will be helpful to achieve good rates.

— Random like will imply good distance.

— Being explicit will make codes explicit.

# Distance Amplification

We illustrate with a (high-level) example.

(i) Start with a base code $C_0$

$[n, r_0, \delta_0]_2$ with $r_0, \delta_0 = \Omega(1)$.

(ii) Take a bipartite expander

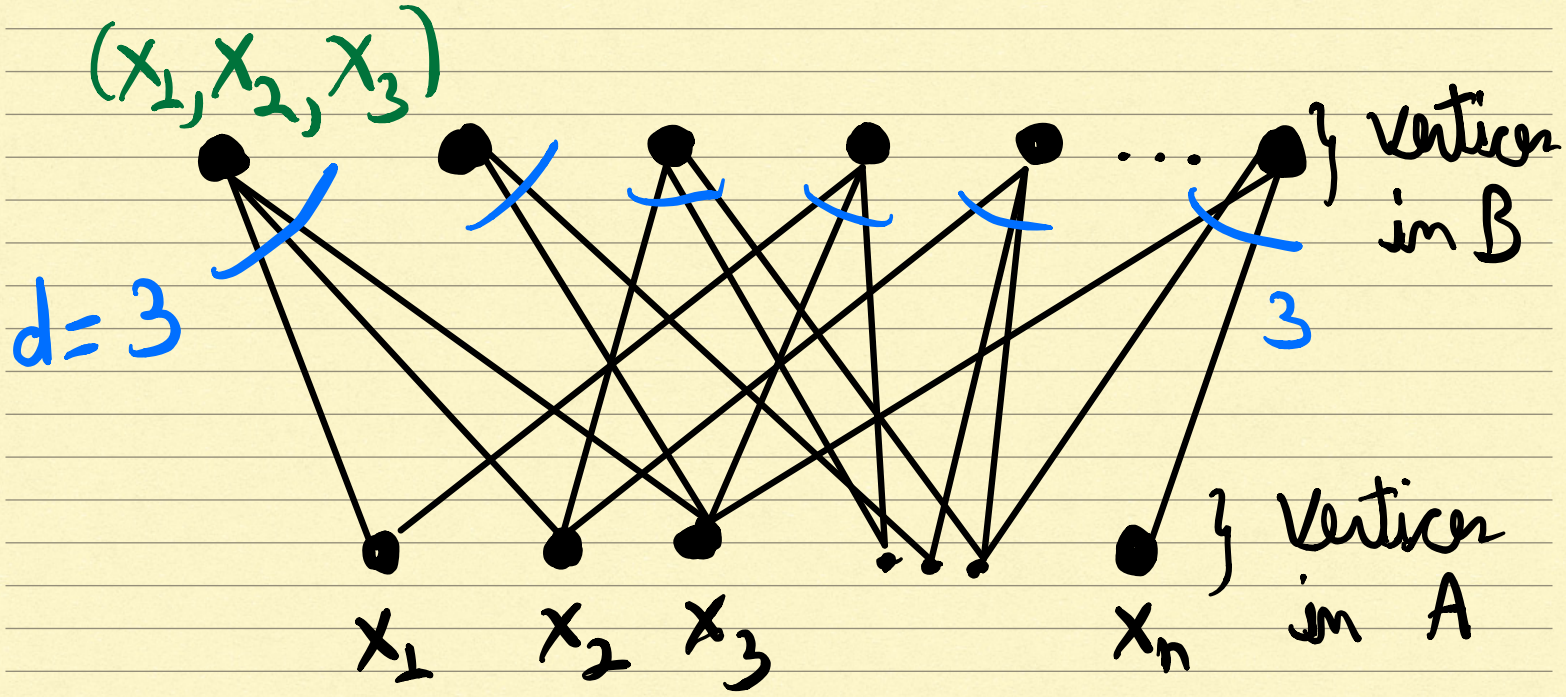
$G = (A \sqcup B, E)$ with

$|A| = n$

$\deg(v) = d \quad \forall \, v \in B.$

(the expansion of $G$ needs to be strong enough)

$x \in \mathbb{F}_2^n$, $G = (A \cup B, E)$




Assume $\forall v \in B$, $N(v) = \{v_1, v_2, v_3\}$.
(i.e., neighbors of $v$ are labeled)

Form a new codeword from $x$ denoted $\text{lift}(x) \in (\mathbb{F}_2^d)^{|B|}$ as:

$$\text{lift}(x)_v := (x_{v_1}, x_{v_2}, x_{v_3})$$

for every $v \in B$.

# Why is the distance amplified?

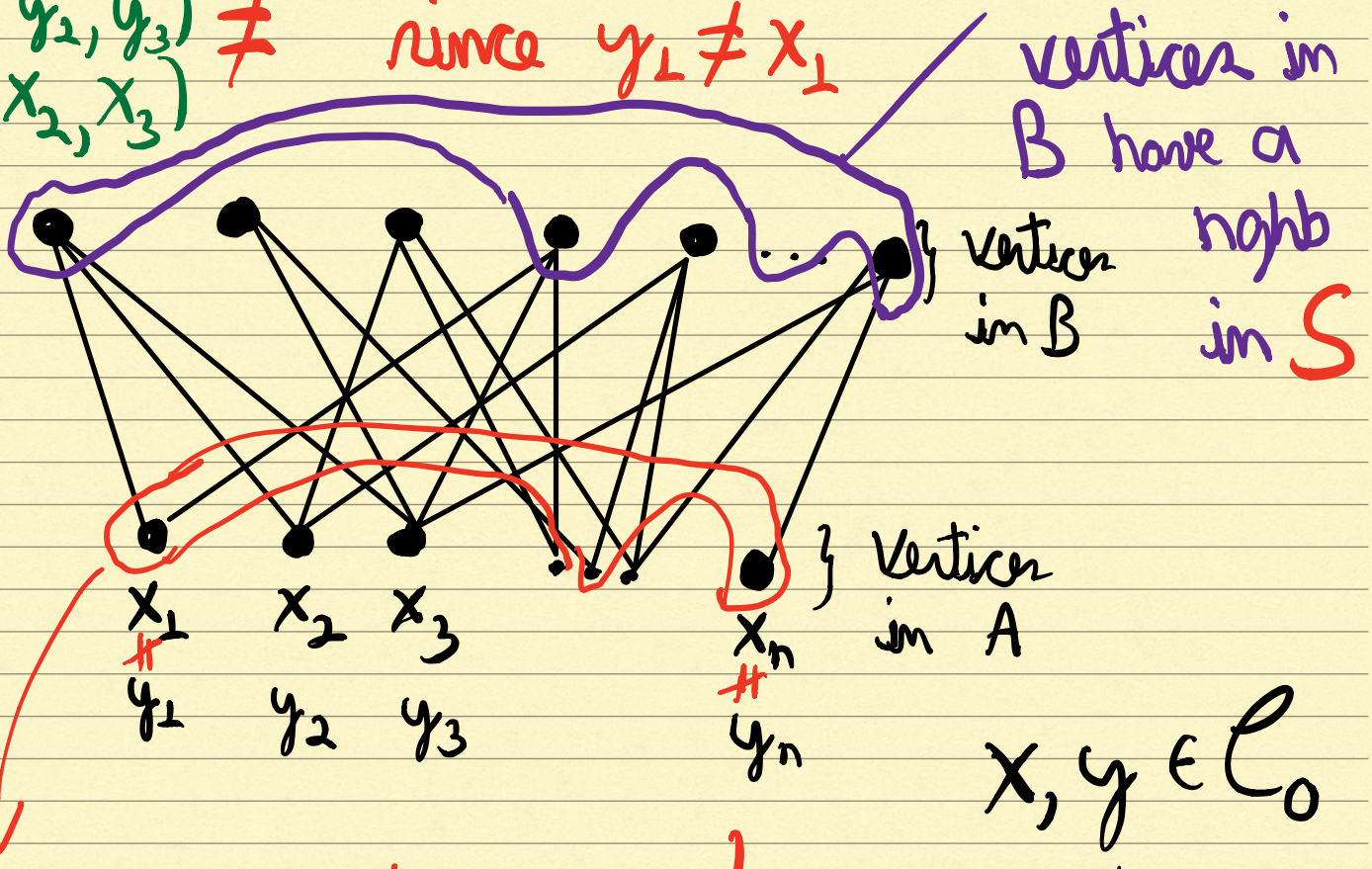
We use the "sampling" property of an expander:

$\forall S \subseteq A$ with $|S| \geq \beta_0 \cdot |A|$

most vertices (depending on $\beta_0$ and the expansion of $G$) of $B$ have neighbors in $S$.

Ex:

$(y_1, y_2, y_3) \neq (x_1, x_2, x_3)$ since $y_1 \neq x_1$




$x, y \in \mathcal{C}_0$

$x \neq y$

$S = \{i \in A \mid x_i \neq y_i\}$

$|S| = \Delta(x, y) \geq \Delta(\mathcal{C}_0)$

This implies that

$$\Delta(\mathrm{lift}(x), \mathrm{lift}(y)) = 1 - \varepsilon,$$

where $\varepsilon = \varepsilon(\lambda, \beta_0)$.

( spectral gap

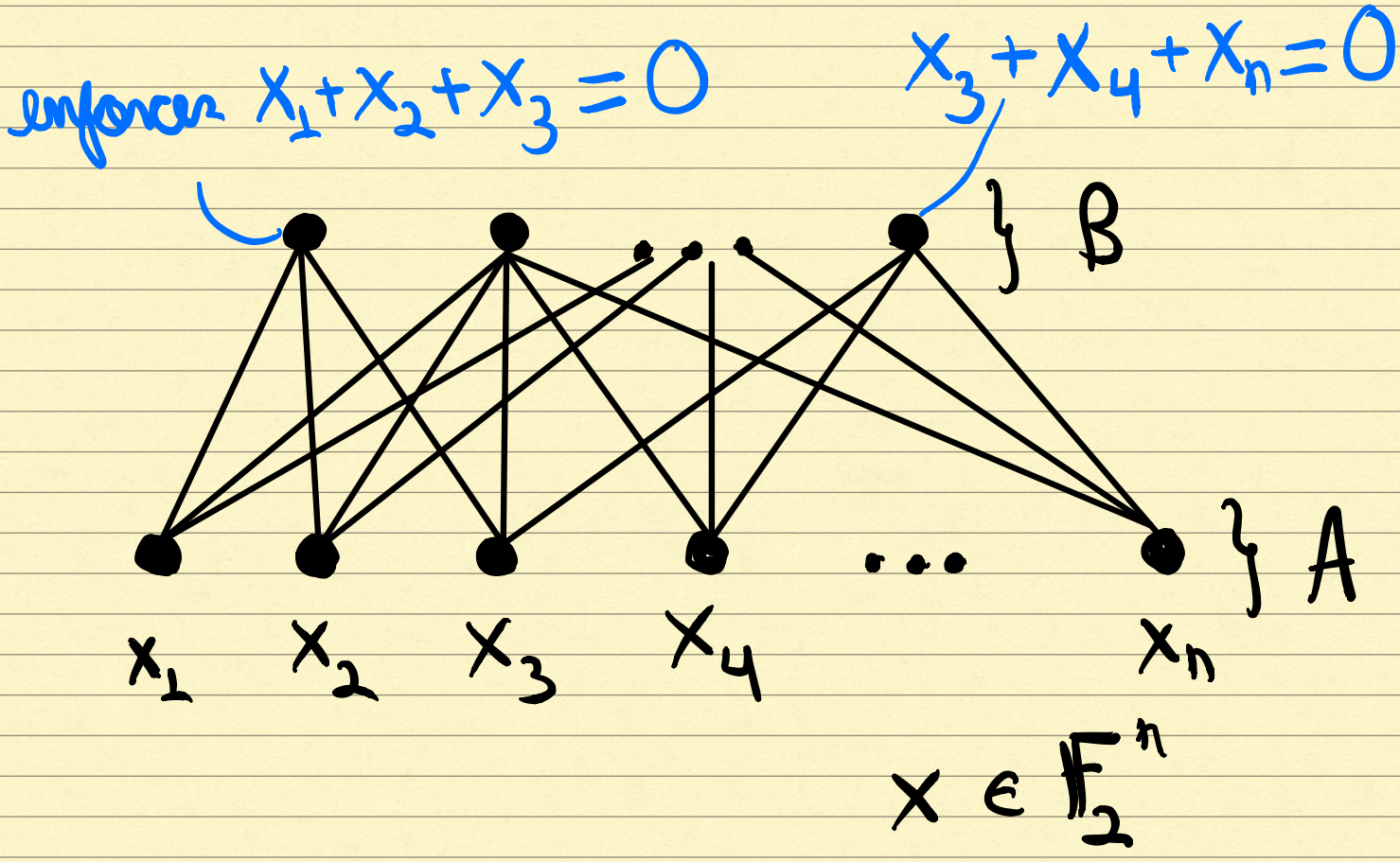
Exercise: Compute bounds on $\varepsilon$.

Note: the specific "lifting" function used in our example is the well-known direct product encoding.

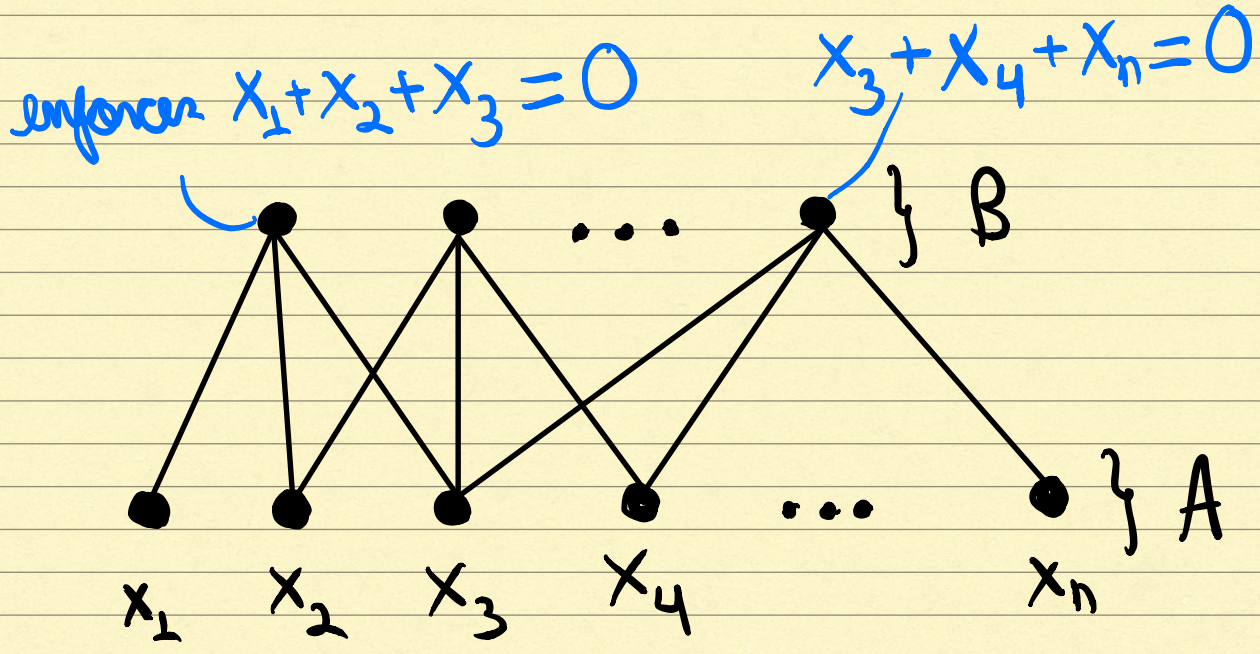
# Expanders and Parity Check

The adjacency relation of a bipartite (vertex) expander is used to define the parity check matrix of a code.

$$G = (A \cup B, E)$$




Require: $|B| \leq \rho |A|$ $\quad \rho \in (0,1)$




Naturally define $H \in \mathbb{F}_2^{B \times A}$ as

$$H_{v,i} := \mathbb{1}_{[v \sim_G i]}$$

to be the parity check matrix of $\mathcal{C}$.

$$\dim(\mathcal{C}) \geq |A| - |B| \geq (1-p)|A|$$
$$= (1-p) \cdot n.$$

# Minimum distance

Exercise: use a sufficiently strong vertex expansion assumption to obtain minimum distance.

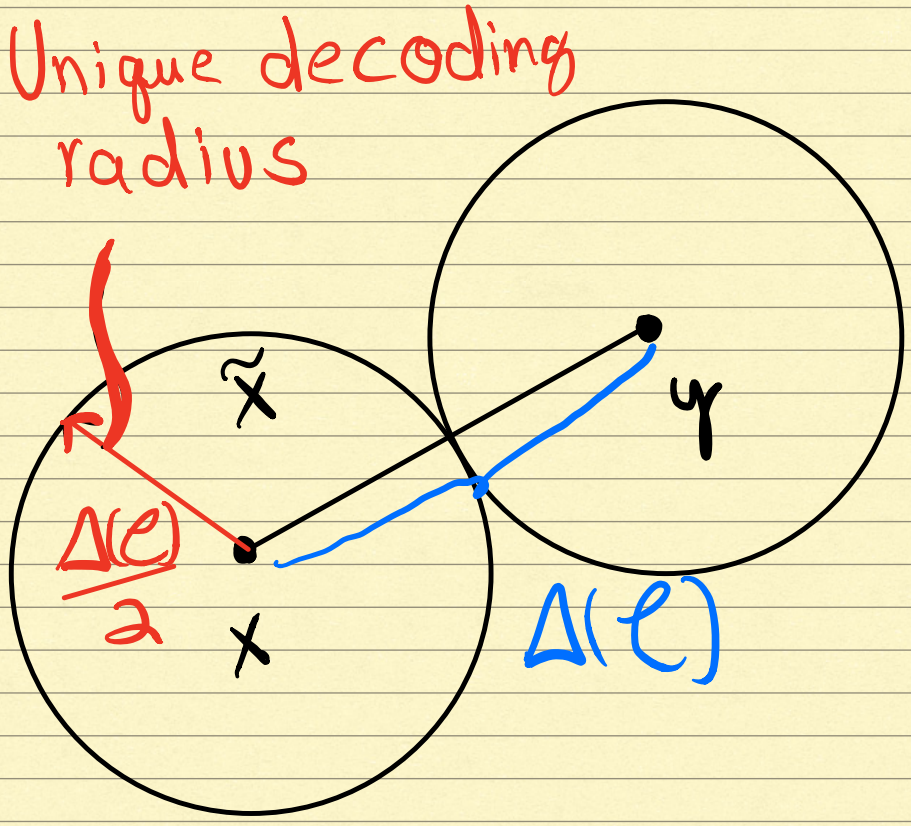
Decoding can be done using a simple local algorithm.

The above construction is due to Sipser and Spielman.
Their code belongs to the more general class of so-called Tanner codes (which allow more general parity constraints).

[See also Guruswami's LDPC survey and Montanari's book.]

# List Decoding

List decoding is a relaxed decoding scheme that allows one to roughly double the unique decoding radius (for well behaved codes) with the caveat of allowing a small list of codewords



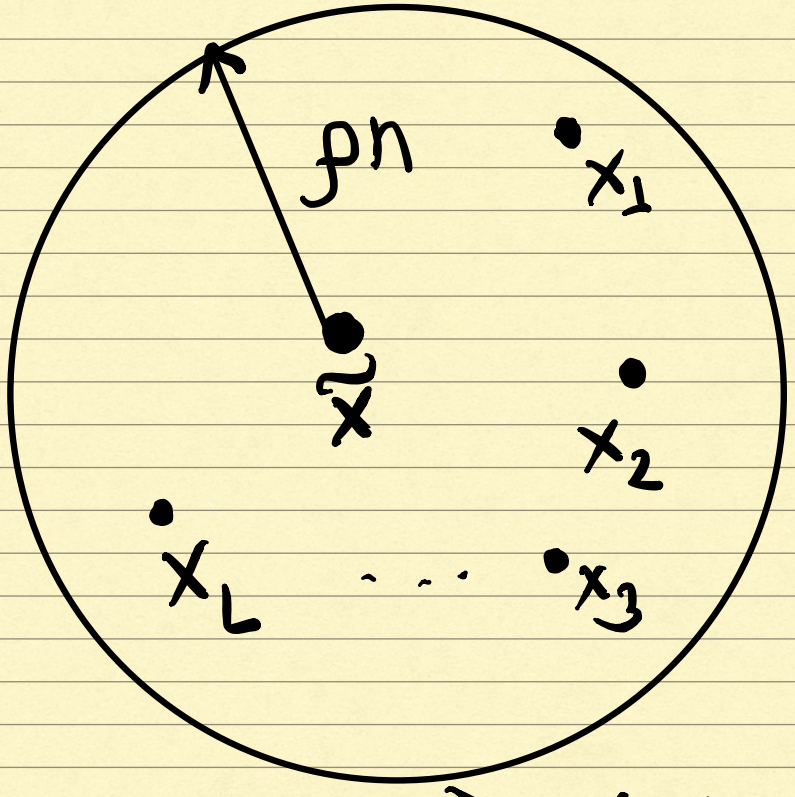
Setting $\mathcal{C} \subseteq \Sigma^n$, $\rho \in (0,1)$

and arbitrary $\tilde{x} \in \Sigma^n$.

Define the list

$$L(\tilde{x}, \rho \cdot n) := \{x \in \mathcal{C} \mid \Delta(x, \tilde{x}) \leq \rho n\}.$$

Ex:




$x_1, \ldots, x_L \in \mathcal{C}$

Possibly $\rho \approx \Delta(\mathcal{C})$ (rather than $\frac{\Delta(\mathcal{C})}{2}$)

Def: $\mathcal{C} \subseteq \Sigma^n$ is $(\rho, L)$-list decodable if $|L(\tilde{x}, \rho n)| \leq L \quad \forall \tilde{x} \in \Sigma^n$.

If $\mathcal{C} \subseteq \Sigma^n$ is $(\rho, L)$-list decodable with $L \leq \text{poly}(n)$, then $\mathcal{C}$ is said to be combinatorially list decodable.

List decoding is an important topic in modern Coding Theory.

# Gilbert-Varshamov Bound

Non-constructive existential bound for codes.

**Theorem: (GV Bound)**

$$(\forall q \geq 2)(\forall p \in (0, 1-\tfrac{1}{q}))(\forall \varepsilon > 0)$$

$$(\exists \text{ a family of } q\text{-ary } \mathcal{C}_n \text{ codes s.t.}$$

(i) $\Delta(\mathcal{C}_n) \geq p$, and

(ii) $r(\mathcal{C}_n) \geq 1 - H_q(p) - \varepsilon)$.

$n$ is the block length

The GV bound follows from the trivial bound:

Fact 1: $\alpha(G) \geq \frac{|V(G)|}{\deg(G)+1}$,

where $G$ is a regular graph of degree $\deg(G)$.

$G$ for us will have $V = [q]^n$ and $x, y \in V$ are connected $\iff \Delta(x,y) < \rho n$

Rmk 1: Any independent set in $G$ is a code of distance $\geq \rho$.

Rmk 2: $\deg(x) = |B(x, \rho n)| \approx q^{H_q(\rho)n}$

Rmk 1 + Rmk 2 + Fact 1

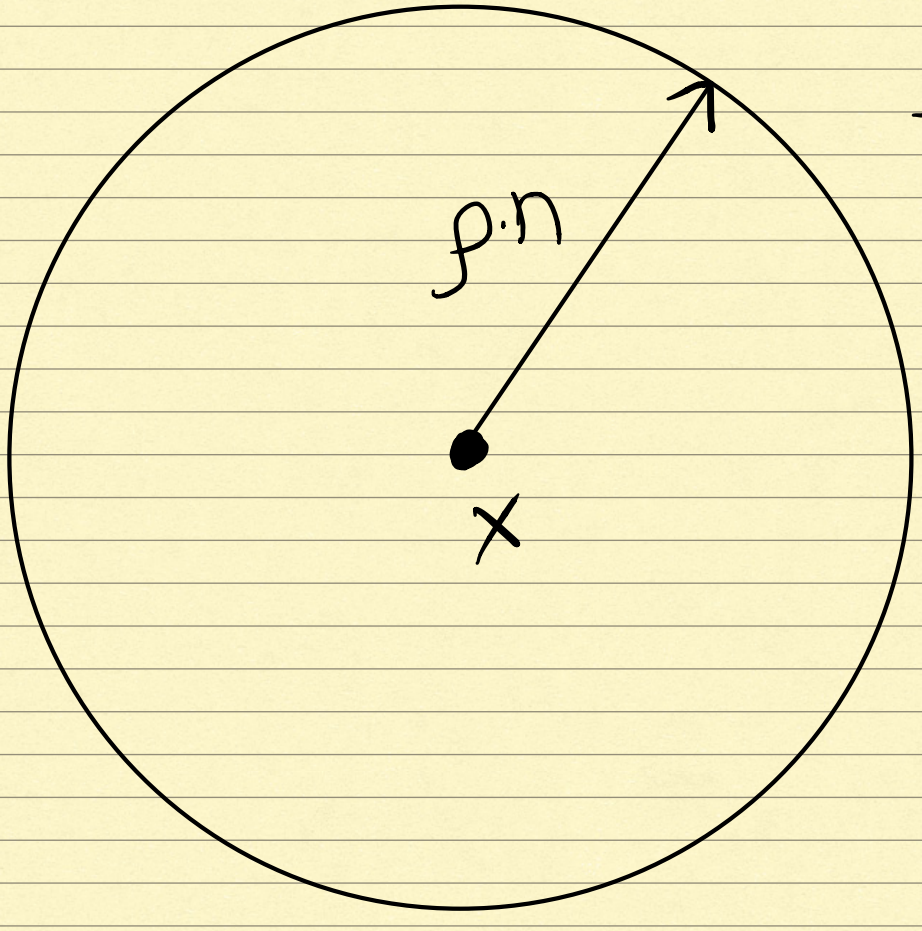
$\Downarrow$

$\exists\, \mathcal{C} \subseteq V$ s.t. $\begin{cases} \Delta(\mathcal{C}) \geq \rho, \text{ and} \\ |\mathcal{C}| \geq \frac{q^n}{q^{H_q(\rho)n}+1} \end{cases}$

i.e., $\Delta(\mathcal{C}) \geq \rho$ and

$$r(\mathcal{C}) \geq 1 - H_q(\rho) - \varepsilon .$$

In a picture, we are covering the $q$-ary $n$-cube with balls of radius $\rho n$.

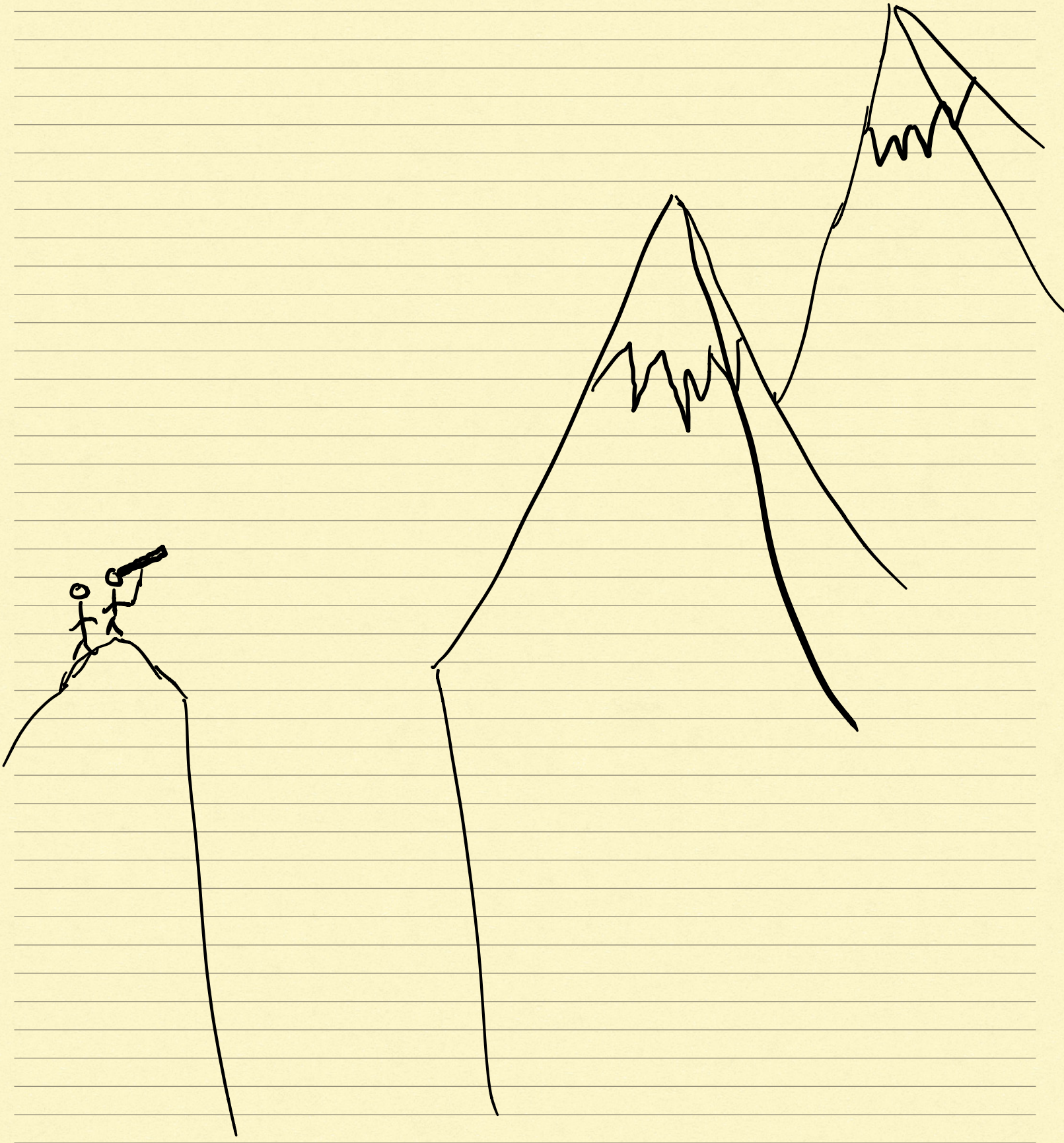



$B(x, \rho n)$
Hamming ball of radius $\rho n$.

Despite its simple form the GV bound is the best known existential bound for binary codes of constant rate and constant relative distance.

(The GV Bound also holds for linear codes.)

# Open Problems



# List Decoding Capacity for Binary Codes

An ambitious problem in this direction.

**Open Problem:** (☕, ☕, ☕, ..., ☕)

Find an explicit family of binary codes of rate $\Omega(\varepsilon^2)$ efficiently list decodable from radius $\frac{1}{2} - \varepsilon$.

The following relaxations are also interesting.

**Open Problem:** (☕, ☕, ☕, ..., ☕)

Find an ~~explicit~~ family of binary codes of rate $\Omega(\varepsilon^2)$ efficiently list decodable from radius $\frac{1}{2} - \varepsilon$.

**Open Problem:** (☕, ☕, ☕, ..., ☕)

Find an explicit family of binary codes of rate $\Omega(\varepsilon^2)$ ~~efficiently~~ list decodable from radius $\frac{1}{2} - \varepsilon$.

combinatorially

# What is Known?

Theorem (Guruswami-Rudra) + ...

**Open Problem:**

Find an explicit family of binary codes of rate $\Omega(\varepsilon^3)$ efficiently list decodable from radius $\frac{1}{2} - \varepsilon$.

Theorem (Guruswami-Rudra) + (Mosheiff et al.) + ...

**Open Problem:**

Find a ~~explicit~~ family of binary codes of rate $\Omega(\varepsilon^2)$ ~~efficiently~~ list decodable from radius $\frac{1}{2} - \varepsilon$.

Combinatorially

# Good LTCs

**Open Problem:** (☕, ☕, ..., ☕)

Find an explicit family of

$\ell$-LTCs with

- Constant number of queries, i.e., $\ell = O(1)$,
- constant rate, and
- constant relative distance.

(This problem is related to the open problem of linear size PCPs)

The following relaxations are also interesting.

**Open Problem: (~~✓~~, ~~✓~~, ..., ~~✓~~)**

Find an ~~explicit~~ family of $\ell$-LTCs with

- Constant number of queries, i.e., $\ell = O(1)$,
- constant rate, and
- constant relative distance.

**Open Problem: (~~✓~~, ~~✓~~, ~~✗~~, ~~✓~~)**

Find an explicit family of $\ell$-LTCs with

- ~~Constant number of queries, i.e., $\ell = O(1)$,~~ improve the dependence $\ell(n)$
- constant rate, and
- constant relative distance.

# What is known?

Theorem (Ben-Sasson and Sudan) + (Dinur) + (Mein)

not entirely explicit

**Open Problem:**

Find an explicit family of $\ell$-LTCs with

- Constant number of queries, i.e., $\ell = O(1)$,
- ~~Constant rate~~, and
- Constant relative distance.

rate $1/\text{polylog}(n)$, where $n$ is the block length

# LTCs and HDX

(a digression)

HDXs constructions can yield linear size $l = O(1)$-uniform hypergraphs with nice properties.

total # hyperedges $= O_l($# vertices$)$.

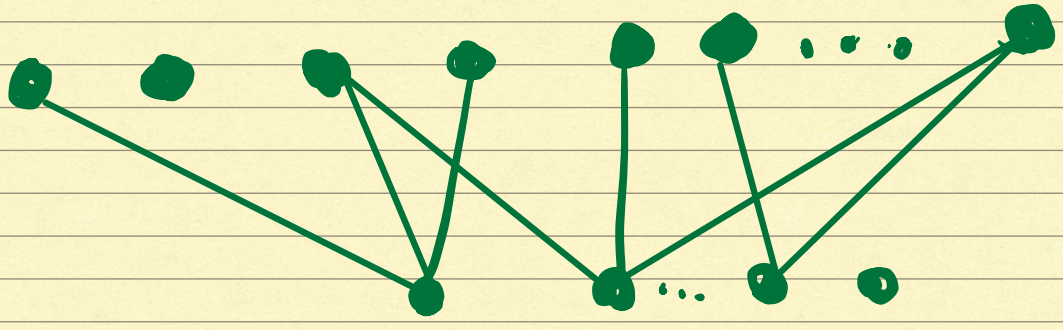
HDX machinery might be helpful towards building good LTCs (according to some specialists).

(borrowing from Mary Wootters)

In this direction, we have:

**Open Problem:**
Are there constant distance, constant rate Tanner codes on MDXs?

Difficulty:




$|B| > |A|$

$H \in \mathbb{F}^{B \times A}$ parity check

naive counting cannot ensure non-trivial rate

Counting is against us

but some algebraic structure

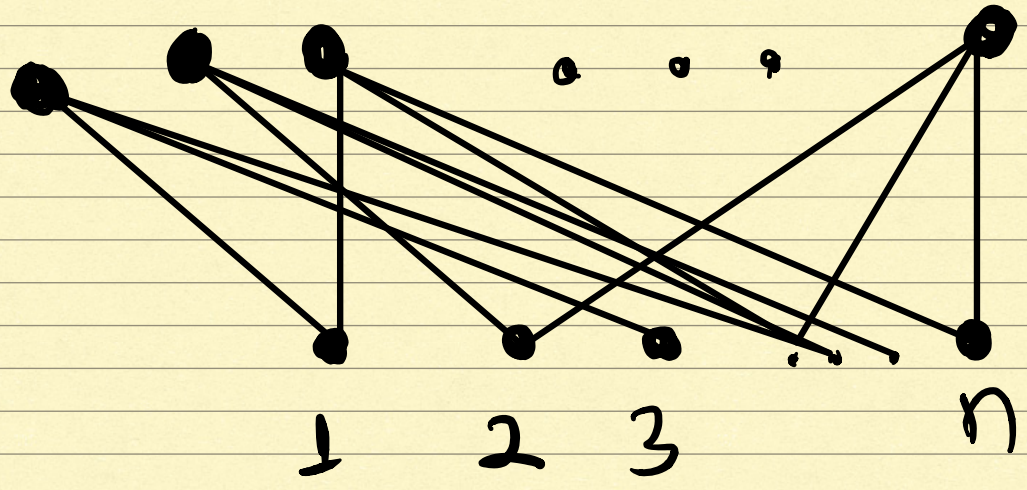
can guarantee a non-trivial kernel

Ex: $\mathrm{Had}_k$ $\quad f_x \quad x \in \mathbb{F}_2^k$

$n = 2^k$ $\qquad \{(x, y, x+y) \mid x, y \in \mathbb{F}_2^k\}$

$=$

$X(3)$




$X(1)$

$=$

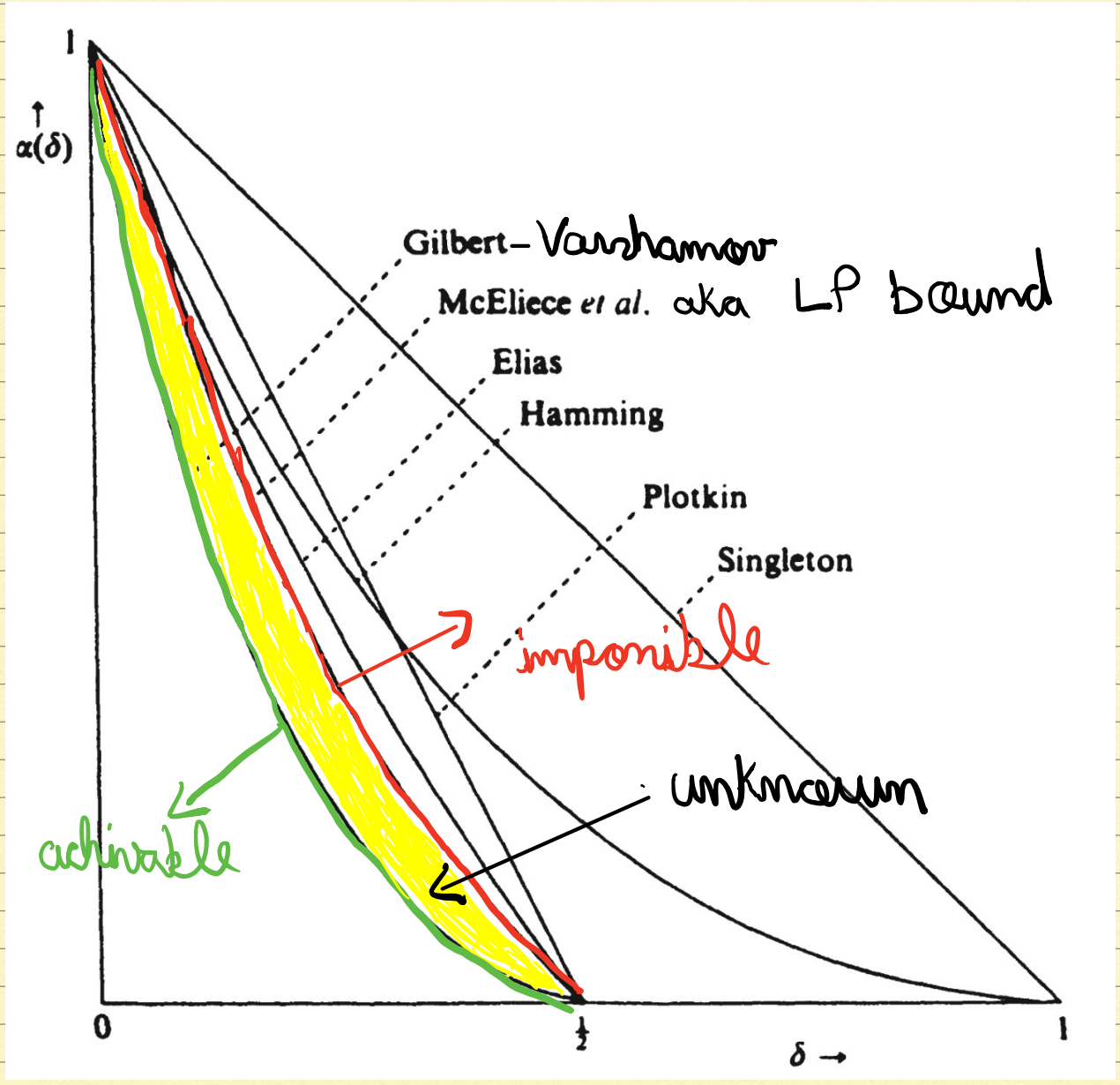
$\mathbb{F}_2^k$

$n^2 = |X(3)| \gg |X(1)| = n$

yet $r(\mathrm{Had}_k) = \log_2 n / n \gg 1/n$.

# Distance vs Rate trade-off for Binary Codes

Noticeable gap between existential and impossibility bounds.




Ref: adapted from van Lint's book.

**Open Problem:** (☕, ☕, ☕, ..., ☕)

Find the best asymptotic trade-off between rate and distance for binary codes.

(This is known only for some cases)

# What is known?

Asymptotically for good binary codes:

- Best lower bound:
  (positive / existential)

  Gilbert-Varshamov Bound
  1950's

- Best upper bound:
  (negative / impossibility)

  LP bounds McEliece et al.
  1970's

# Tighten the Gap between upper and lower Bounds for LDCs

The rate lower bounds for LDCs and the rate of known constructions are very far apart.

**Open Problem:**

Tighten the huge rate gap of constant query LDCs.

# To give an idea of the gap

(this may be a bit outdated)

$k/n$ is the rate with $n$ being the block length

**Theorem** (Efremenko)

$\exists$ 3-LDCs with

$$n = \exp(\exp(\sqrt{\log k \log\log k}))$$

**Theorem** (Woodruff)

$\forall$ 3-LDCs $n = \Omega(k^2)$.

(linear)

(see Yekhanin's survey)

# "Zig-Zag product" of High-dimensional Expanders

The zig-zag product of Reingold, Vadhan and Wigderson is an operation that combines two expander graphs $G$ and $H$ to produce a larger one $G \circledcirc H$. It has several applications including to coding theory.

**Open Problem:**

Given two MDXs (spectral link expanders) $X$ and $Y$, can we produce a larger one $X \text{“}\circledcirc\text{”} Y$?

# Improve the Running Time of a Unique Decoder of explicit $\varepsilon$-balanced codes near the GV Bound

We now have polynomial time decoding for these codes, but nowhere near quadratic or near-linear.

**Open Problem:**

Improve the decoder running time to near-linear time.

# Lower Bounds for Average Case Coding Tasks against the Sum-of-Squares Hierarchy

A sample question in this direction (probably requires more polishing/thinking)

**Question:**

Can we fool SOS into thinking that a random linear code (given by a random generator matrix) has a codeword of small Hamming weight?

I would not classify it as open at this point.

That's all.

Thank you!