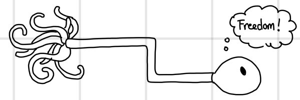
CS 250/EE 387 - LECTURE 3 - MORE LINEAR CODES & APPLICATIONS to CRYPTO, & ASYMPTOTICS!

AGENDA

1. GV Bound
2. Efficiency?
3. Detour: McEliece crypto system
4. Off to asymptopia!
5. Q-ary entropy

TODAY'S OCTOPUS FACT

Octopuses have no bones, and can squeeze through very small spaces. In 2016, an octopus named "Inky" in a New Zealand aquarium escaped to the ocean through a 6-inch diameter drainpipe.



⓪ Recap.

Last time, we saw LINEAR CODES:

$$C = \left\{ \underbrace{\begin{bmatrix} \\ G \\ \\ \end{bmatrix}}_{n \times k} \begin{bmatrix} \\ \\ \end{bmatrix}_x : x \in \mathbb{F}_q^k \right\} = \left\{ c \in \mathbb{F}_q^n : \begin{bmatrix} \quad H \quad \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix}_c = 0 \right\}$$

G is a GENERATOR MATRIX

H is a PARITY-CHECK MATRIX.

- Linear algebra works great if $\mathbb{F}_q$ is a field.
- This is super useful.

Today: A few useful things one can do with linear codes:

- GILBERT-VARSHAMOV BOUND
- McELIECE CRYPTOSYSTEM. Technically, this is a useful thing you CAN'T do with linear codes.

(Plus, a few combinatorial things you can't do with any codes, useful or otherwise).

1 The GILBERT VARSHAMOV BOUND

So far, we have seen the **HAMMING BOUND**, which is an upper bound on the rate of a code. (aka, an **IMPOSSIBILITY RESULT**). We can match it for $n=7$, $k=4$, but what about in general?

Next, we'll see the **GILBERT-VARSHAMOV BOUND**, which is a **POSSIBILITY RESULT**.

THM (GILBERT-VARSHAMOV)

For any prime power q, and for any $d \leq n$, there exists a linear code C of length n, alphabet size q, distance d, and rate

$$R \geq 1 - \frac{\log_q(\mathrm{Vol}_q(d-1, n)) - 1}{n}$$

NOTE: You can remove the words "prime power" and "linear" and the statement is still true.

Compare this to the Hamming bound, which said:

$$R \leq 1 - \frac{\log_q\left(\mathrm{Vol}_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right)\right)}{n}$$

The only difference is this part.

We will talk more about the relationship between these two later, but for now just notice that $R_{GV} < R_{HAMMING}$, so math is not broken.

We'll prove the GV bound now – it's pretty easy! However, it's **NOT KNOWN** if we can do better in general!

OPEN **QUESTION** Do there exist binary codes that do better than the GV bound? (for all parameter regimes?)

Proof of the GV bound.

This is called the PROBABILISTIC METHOD.

IDEA: A <u>random</u> linear code will do the trick with probability >0.
So, in particular, there exists a linear code that works!

Let C be a random subspace of $\mathbb{F}_q^n$, of dimension k. (k TBD)
Let G be a random generator matrix for C.

USEFUL FACT: For any fixed $x \neq 0$, $G \cdot x$ is uniformly random in $\mathbb{F}_q^n \setminus \{0\}$.

Informal proof: Because of all the symmetry, how could it be any other way?
Formal proof: Fun exercise!

Now, $\mathrm{dist}(C) = \min_{c \in C \setminus \{0\}} \mathrm{wt}(c) = \min_{x \in \mathbb{F}_q^k \setminus \{0\}} \mathrm{wt}(G \cdot x)$

For any given $x \neq 0$, by the USEFUL FACT,

$$\mathbb{P}_G\left\{ \mathrm{wt}(G \cdot x) < d \right\} = \mathbb{P}_G\left\{ G \cdot x \in B_q(0, d-1) \right\}$$

$$\leq \frac{\mathrm{Vol}_q(d-1, n)}{q^n}$$

number of choices for $G \cdot x$ w/ wt $\leq d-1$

number of choices for $G \cdot x$ total.

(Technically it's $\frac{\mathrm{Vol}(d-1,n)-1}{q^n - 1}$, which is $\leq$ the above)

So by the union bound,

$$\mathbb{P}\left\{ \exists x \in \mathbb{F}_q^k : \mathrm{wt}(G \cdot x) < d \right\} \leq q^k \cdot \frac{\mathrm{Vol}_q(d-1, n)}{q^n}.$$

Thus, we win as long as this is $\lneq 1$. Taking $\log_q$ of both sides, we win if

$$k - n + \log_q(\mathrm{Vol}_q(d-1, n)) < 0$$

So choose $k = n - \log_q(\mathrm{Vol}_q(d-1, n)) - 1$, and we are done. ∎

② EFFICIENCY (?)

- If C is linear, we have an efficient encoding map $x \mapsto G \cdot x$

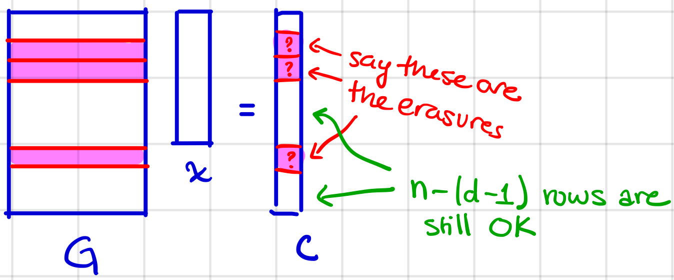
  The computational cost is one matrix-vector multiply

- If C is linear with distance d, we can DETECT $\leq d-1$ errors efficiently:

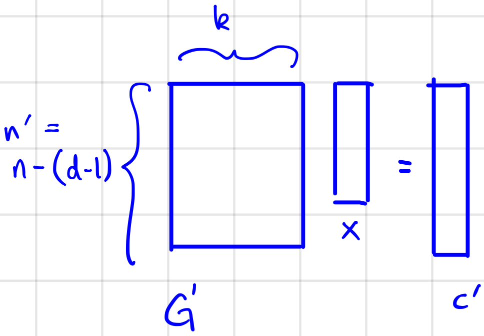
  If $0 < wt(e) \leq d-1$ and $c \in C$, then $H(c+e) = H \cdot e \neq 0$, so just check if $H\tilde{c} \neq 0$.

- If C is linear with distance d, we can CORRECT $\leq d-1$ ERASURES efficiently:

  We have



$\Rightarrow$ get rid of the erased rows



Because the distance is d, there is exactly one $c' \in \mathbb{F}^{n'}$ that is consistent with these equations, and hence G' is full rank.

$\Rightarrow$ Solve this linear system $G'x = c'$ for x.

- If C is linear with distance d, can we CORRECT $\lfloor \frac{d-1}{2} \rfloor$ ERRORS efficiently?

  - It worked for the $(7,4,3)_2$-Hadamard code!
  - But what about in general?

ASIDE: Can we still solve linear systems efficiently over finite fields? Sure!

All your favorite algorithms (eg, Gaussian Elimination) only need addition, subtraction, multiplication and division, so that still works over $\mathbb{F}_q$.

- Consider the following problem:

$$\boxed{\text{Given } \tilde{c} \in \mathbb{F}_q^n, \text{ and } G \in \mathbb{F}_q^{n \times k}, \text{ find } x \in \mathbb{F}_q^k \text{ s.t. } \Delta(G \cdot x, \tilde{c}) \text{ is minimized.}}$$

aka, find the codeword closest to a received word $\tilde{c}$.

- This problem (called MAXIMUM-LIKELIHOOD DECODING for LINEAR CODES) is NP-hard in general [Berlekamp-McEliece-van Tilborg 1978], even if the code is known in advance and you have an arbitrary amount of preprocessing time [Bruck-Noar 1990, Lobstein 1990]. It's even NP-hard to approximate (within a constant factor)! [Arora-Babai-Stern-Sweedyk 1993].

- Even computing the minimum distance of linear codes is NP hard!

- This all sounds bad, but remember that NP-hardness is a worst-case condition. There exist linear codes that are (probably) hard to decode, but that doesn't mean that all of them are.

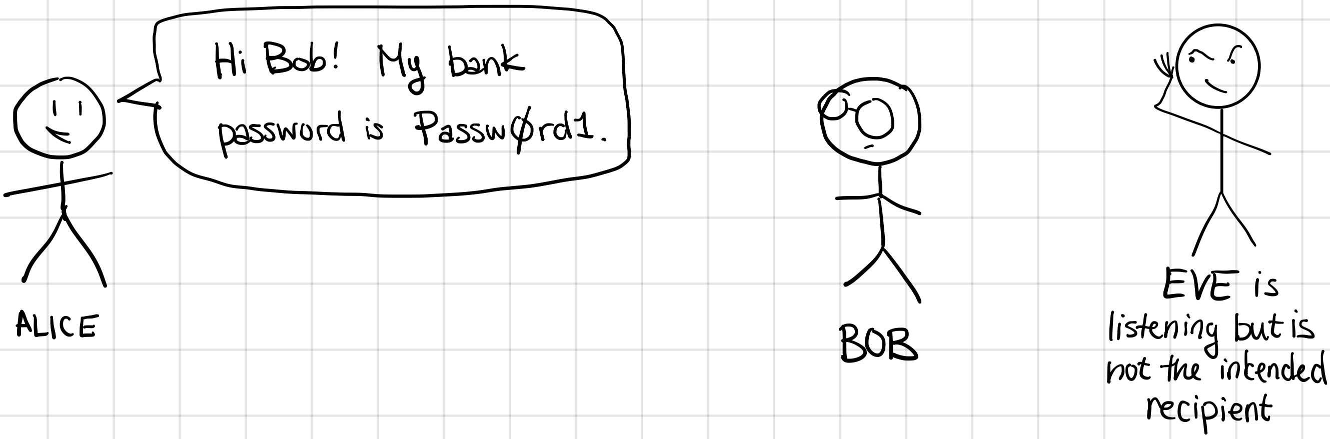
- We will spend most of this class talking about how to get efficiently-decodable codes. But first let's see one application where the hardness is a good thing.

NOTE: Actually, it's not really clear what "NP hard" or "computational efficiency" means here. (Besides the fact that we did not define them).
In particular, these notions make sense as the input size grows.
What is growing? We'll come back to this in a moment.

\begin{detour}

③ Application: McEliece Cryptosystem

Suppose that Alice and Bob want to talk SECURELY.
Now there is no noise, just an EAVESDROPPER Eve.



- In **PUBLIC KEY CRYPTOGRAPHY**, everyone has a public key and a private key.
- To send a message to Bob, Alice encrypts it using Bob's public key.
- Bob decodes it with his private key.
- We hope that this is secure as long as Bob's private key stays private.

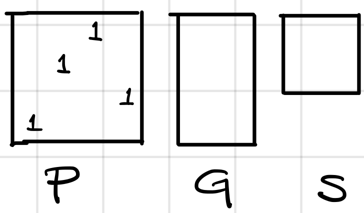
Here is such a scheme, using binary linear codes:

- Bob chooses: $G \in \mathbb{F}_2^{n \times k}$ is the generator matrix for an (appropriate) efficiently decodable binary linear code $\mathcal{C}$.
  from t errors

  The McEliece system uses something called a "Goppa Code", but we will not go into details in these notes.

- Bob chooses:
  - A random invertible $S \in \mathbb{F}_2^{k \times k}$
  - A random permutation matrix $P \in \mathbb{F}_2^{n \times n}$

- Bob's private key: (S, G, P)

- Bob's public key: $\hat{G} =$  , and t

Scheme continued...

To send a message $x \in \mathbb{F}_2^k$ to Bob:

- Alice chooses a random vector $e \in \mathbb{F}_2^n$ with $wt(e) = t$
- Alice sends Bob $\hat{G}x + e$

To decrypt Alice's message $\hat{G}x + e$:

- Bob computes $P^{-1}(\hat{G}x + e) = GSx + P^{-1}e = GSx + e'$, where $wt(e') = t$
- Bob uses his efficient decoder for C to find $S \cdot x$
- Bob computes $x = S^{-1} \cdot Sx$

Why might this be secure?

Suppose Eve sees $\hat{G}x + e$.
She knows $\hat{G}$ and t, so this problem is the same as decoding the code $\hat{C} = \{\hat{G}x \mid x \in \mathbb{F}_2^k\}$ from t errors. WE HOPE THIS IS HARD.

Note: "Decoding $\hat{G}x + e$ is hard for Eve" is NOT the same as "Maximum likelihood decoding of linear codes is NP-hard."

- First, we have some promise that there were $\leq t$ errors.
- Second, NP-hardness is a worst-case assumption: for crypto we need an average-case assm.

The assm that "Decoding $\hat{G}x + e$ is hard" (for an appropriate choice of G) is called the McELIECE ASSUMPTION. Some people believe it and some don't.

\end{detour}

④ OFF TO ASYMPTOPIA

We'll return to computational issues later — but first we need to talk about what we mean by "for large n". So for now let's return to the combinatorial question:

WHAT IS THE BEST TRADE-OFF between RATE and DISTANCE?

So far we've seen two bounds:

Gilbert Varshamov

Hamming

$$1 - \frac{1}{n} \cdot \log_q\left(\mathrm{Vol}_q(d-1, n)\right) - \frac{1}{n} \leq \text{optimal } k/n \leq 1 - \frac{1}{n} \cdot \log_q\left(\mathrm{Vol}_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right)\right)$$

for $|\Sigma| = q$

So for particular d, k, n, these tell us something ... but what do they tell us in general? And what does "in general" mean?

We are going to think about the following limiting parameter regime:

$$\boxed{n, k, d \longrightarrow \infty \quad \text{so that} \quad \frac{k}{n} \to R \quad \text{and} \quad \frac{d}{n} \to \delta \quad \text{approach constants.}}$$

Motivations:

(1) It will allow us to better understand what's possible and what's not

(2) In many applications, n, k, d are pretty large and R, δ are the things we want to be thinking about.

(3) It will let us talk meaningfully/rigorously about computational complexity.

DEF A **FAMILY of CODES** is a collection $\mathcal{C} = \{ \mathcal{C}_i \}_{i=1}^{\infty}$, where $\mathcal{C}_i$ is an $(n_i, k_i, d_i)_{q_i}$ code.

The **RATE** of $\mathcal{C}$ is $\quad R(\mathcal{C}) := \lim_{i \to \infty} k_i / n_i$

The **RELATIVE DISTANCE** of $\mathcal{C}$ is $\quad \delta(\mathcal{C}) := \lim_{i \to \infty} d_i / n_i$

NOTES:

- We will frequently abuse notation and refer to $\mathcal{C}$ as a "code," and we'll drop the subscript i and just think about $n, k, d \to \infty$.
- The alphabet of $\mathcal{C}_i$ might depend on i. (But if it doesn't will say the whole family has alphabet size q).

EXAMPLE. In the in-class exercises, you will investigate HAMMING CODES.
The i^{th} code $\mathcal{C}_i$ is a $(2^i - 1, \ 2^i - i - 1, \ 3)_2$ code.
$\mathcal{C}_i$ is defined by its parity-check matrix,

$$H_i = \left.\overbrace{\begin{bmatrix} | & | & | & | & | & \cdots & | & | & | \end{bmatrix}}^{2^i - 1}\right\} i$$

This matrix has every nonzero binary vector as a column.

The RATE of this family is $\displaystyle \lim_{i \to \infty} \frac{2^i - i - 1}{2^i - 1} = 1.$

The RELATIVE DISTANCE is $\displaystyle \lim_{i \to \infty} 3 / 2^i - 1 = 0.$

Our Question: What is the best trade-off between $R(\mathcal{C})$ and $\delta(\mathcal{C})$?
(in the asymptotic setting)

Easier question: Can we obtain codes with $R(\mathcal{C}) \gneq 0$ and $\delta(\mathcal{C}) \gneq 0$?

DEF. Such a code (with $R(\mathcal{C}) > 0$, $\delta(\mathcal{C}) > 0$) is called **ASYMPTOTICALLY GOOD**.

(5) q-ARY ENTROPY

$$\underbrace{1 - \tfrac{1}{n}\cdot \log_q(\mathrm{Vol}_q(d-1,n))}_{\text{Gilbert Varshamov}} \leq \underset{\text{for } |\Sigma| = q}{\text{optimal } k/n} \leq \underbrace{1 - \tfrac{1}{n}\cdot \log_q\left(\mathrm{Vol}_q\left(\lfloor\tfrac{d-1}{2}\rfloor, n\right)\right)}_{\text{Hamming}}$$

Now that we have an asymptotic parameter regime, how should we parse the GV and Hamming bounds?

In particular, what is $\frac{1}{n}\log_q\big(\underbrace{\mathrm{Vol}_q\left(\lfloor\frac{d-1}{2}\rfloor, n\right)}\big)$ in terms of δ, if $\delta = d/n$?

OK, so this is $\sum_{j=0}^{\lfloor\frac{d-1}{2}\rfloor} \binom{n}{j}(q-1)^j$,
but that is not very helpful.

DEF. The q-ary entropy function $H_q : [0,1] \to [0,1]$ is defined by

$$H_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x).$$

This generalizes the BINARY ENTROPY FUNCTION $H_2(x) = H(x)$, which you may have seen.
The reason we care (for this class) is that $H_q(x)$ captures $\mathrm{Vol}_q(xn)$ nicely:

PROP. Let $q \geq 2$ be an integer, and let $0 \leq p \leq 1 - 1/q$. Then

(i) $\mathrm{Vol}_q(n, pn) \leq q^{n\cdot H_q(p)}$

(ii) $\mathrm{Vol}_q(n, pn) \geq q^{n\cdot H_q(p) - o(n)}$

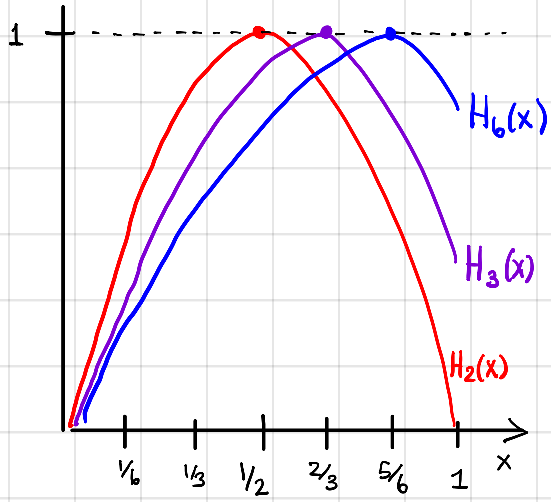
A function $f(n)$ is $o(n)$ if $\frac{f(n)}{n} \to 0$ as $n \to \infty$

See ESSENTIAL CODING THEORY, Chapter 3 for a proof.
[Proof idea: mess around with the binomial coefficients and use Stirling's formula.]

ASIDE: You may have see $H_2(p)$ described in terms of the #bits it takes to describe something. That is, I can describe a random string of length n where each bit is 1 w/ probability p pretty reliably using $H_2(p)$ bits.

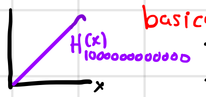
There is a similar interpretation for q-ary entropy. Suppose you choose $x \in \mathbb{F}_q^n$ s.t. each $x_i = \begin{cases} 0 & \text{prob } 1-p \\ \text{random in } \mathbb{F}_q^* & \text{prob } p \end{cases}$
Then the number of bits you need to describe x is roughly $H_q(p)$.

$H_q(x)$ looks ~~like~~ this:
something



Some useful properties [You can see these by taking Taylor expansions]:

- If q is really big (p is reasonable and), then $H_q(p) = p \cdot \underbrace{\log_q(q-1)}_{\text{basically } 1} + \underbrace{\log_q(\text{stuff}) + \log_q(\text{stuff})}_{\text{really small}} \approx p$

  (so, eventually the plot looks like )

- If q is reasonable and p is really small, then $H_q(p) = \underbrace{p \cdot \log_q(q-1)}_{=O(p)} + \underbrace{p\log_q(1/p)}_{\text{This term is the largest}} + \underbrace{(1-p)\log_q\left(\frac{1}{1-p}\right)}_{\approx p/\ln(q) = O(p)}$

  $$\approx p\log_q(1/p)$$

  (so, near 0 all those curves look like $\frac{x\ln(1/x)}{\ln(q)}$)

Now, we can take limits in the GV and Hamming bounds to obtain:

Thm (HAMMING BOUND) For any family $\mathcal{C}$ of q-ary codes,

$$\mathcal{R}(\mathcal{C}) \le 1 - H_q(\delta(\mathcal{C})/2)$$

Proof: follows by taking limits in the Hamming bound.

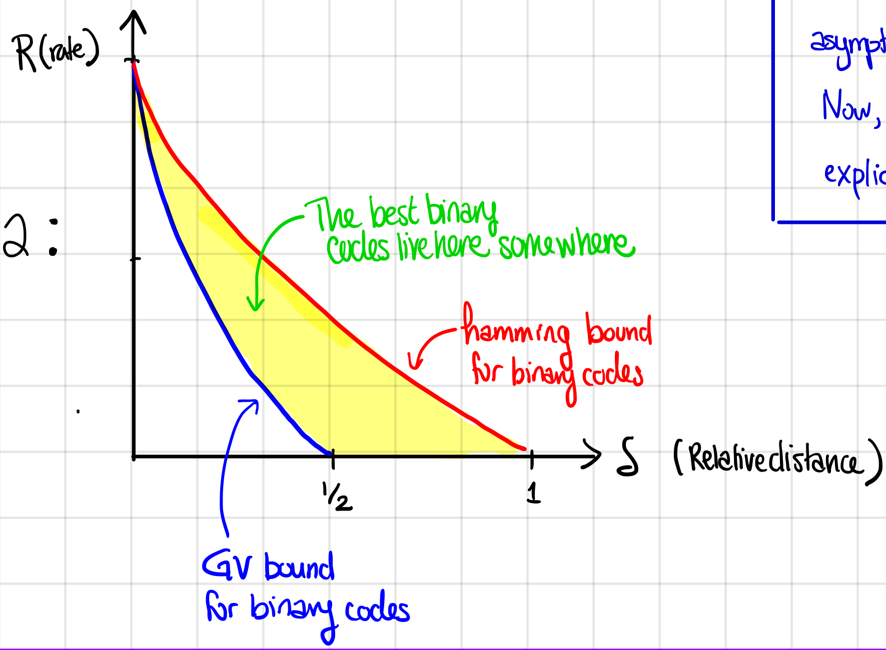
Thm (GV BOUND) Let $q \ge 2$. For any $0 \le \delta \le 1-\frac{1}{q}$, and for any $0 < \varepsilon \le 1 - H_q(\delta)$, there exists a q-ary family of codes $\mathcal{C}$ w/ $\delta(\mathcal{C}) \ge \delta$ and

$$R(\mathcal{C}) \ge 1 - H_q(\delta) - \varepsilon.$$

Proof: FUN EXERCISE (given the nonasymptotic version)

Now it's easier to compare these two bounds.

This is the picture for $q=2$:



Notice: this answers our earlier question. There do exist asymptotically good codes! Now, can we find some explicit ones with efficient algs??

SOME QUESTIONS.

QUESTION Are there families of codes that beat the GV bound?

ANSWER 1: Yes. For $q \geq 49$, "Algebraic Geometry Codes" beat the GV bound.

ANSWER 2: ??? For binary codes, we don't know. OPEN PROBLEM!

QUESTION Can we find explicit constructions of families of codes that meet the GV bound?

ANSWER 1. For large alphabets, yes. (We'll see soon)

ANSWER 2. ??? For binary codes, recent work of [Ta-Shma 2017] gives something close in a very particular parameter regime ... but in general, OPEN PROBLEM!

Questions to Ponder

1. Can you think of strategies to improve the Hamming bound?

2. Is it possible to have a family of codes with $\delta > 1 - 1/q$ and $R > 0$?