

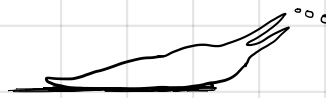
CS250/EE386 - LECTURE 10 - LIST DECODING!

AGENDA

- ① RECAP on SHANNON'S THM
- ① LIST DECODING
- ② LIST DECODING CAPACITY
- ③ JOHNSON BOUND

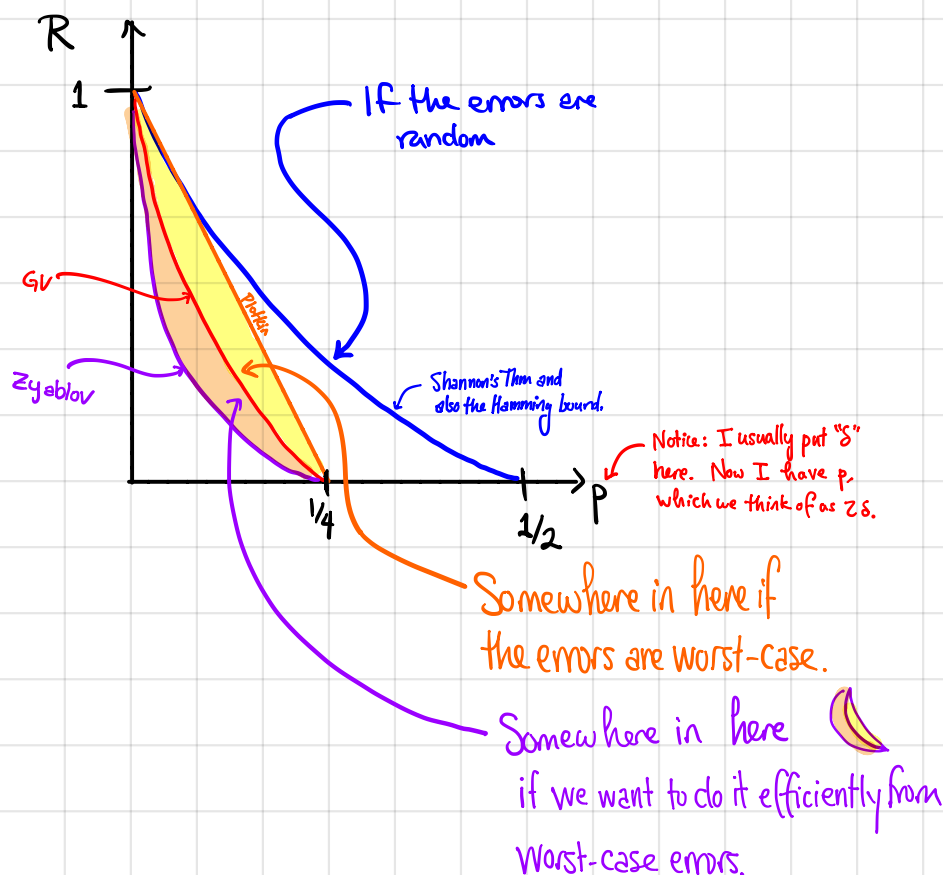
GASTROPOD FACT

Slugs don't have teeth, but they have something called a **RADULA** which has thousands of tiny tooth-like protrusions to help them grind up food.



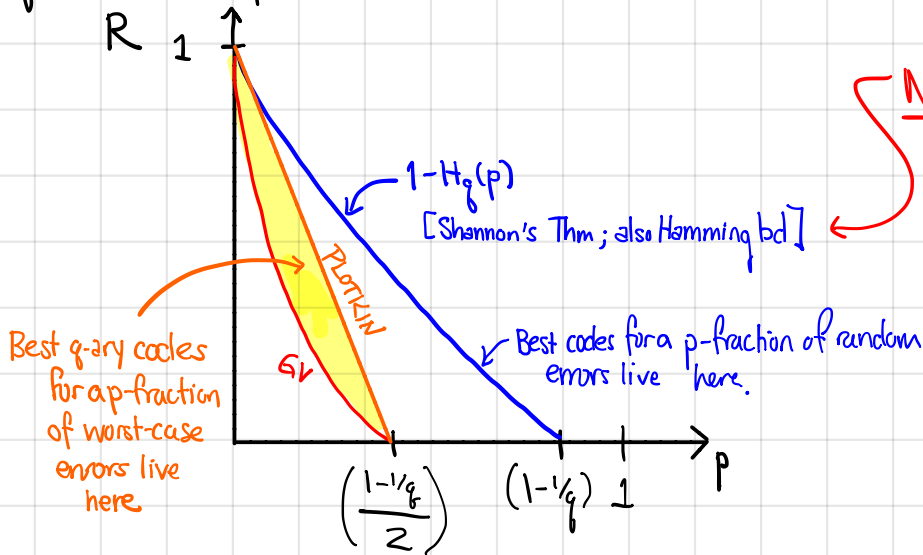
Mother always said not to chew with your mouth open, but for me it's physically impossible to do otherwise!

① The Story So Far: last time we had this graph.



That is, if I want to handle random errors, I can handle **WAY MORE** than if the errors were adversarial!

As $q \rightarrow \infty$, the picture looks similar:



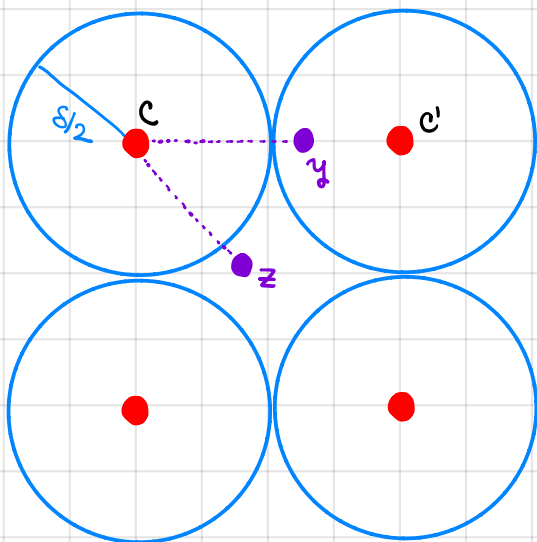
NOTE: We only stated Shannon's Thm for the BSC_p, but it also holds for the q SC_p (the " q -ary symmetric channel")* with $1 - H_q(p)$ instead of $1 - H_2(p)$.

*The q -ary symmetric channel q SC(p) is given by:
 $IP\{x|y\} = \begin{cases} 1-p & x=y \\ p/(q-1) & \text{else} \end{cases}$

we have this really big gap for low-rate codes.
 Almost 100% of random errors is cool, but 50% adversarial errors is not cool.

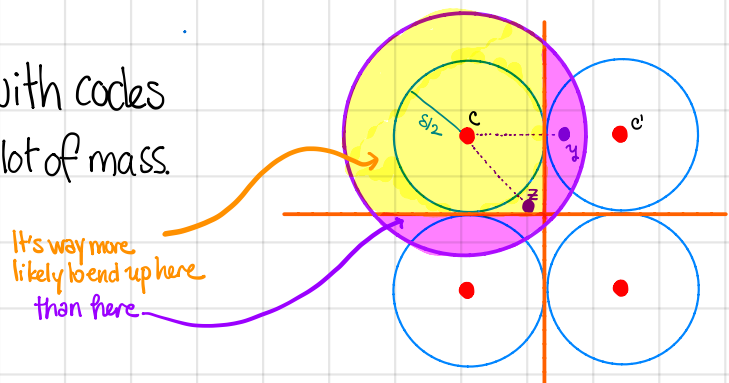
WHY IS THERE SUCH A BIG DIFFERENCE?

Here is a geometric explanation:



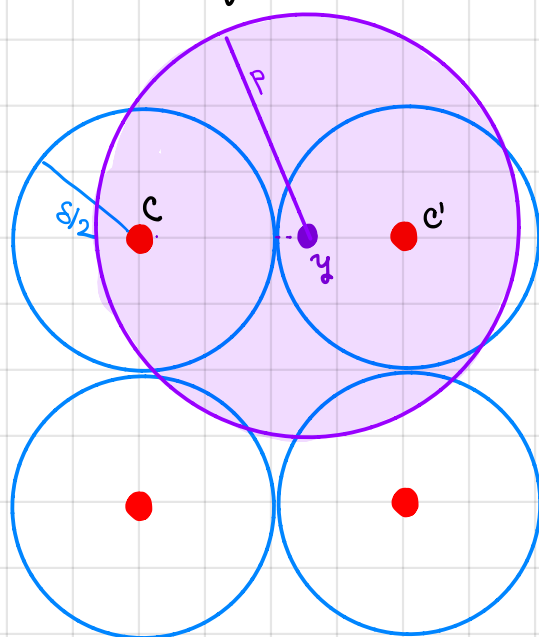
- Suppose c is the "correct" codeword, and there is $> s/2$ error.
- If the errors are adversarial, the adversary might choose y , which would confuse us.
- However, if the errors are random, then z is just as likely as y , and in fact c still is the closest codeword to z . So that would be fine!

The intuition is that we can come up with codes so that these "in between" spaces have a lot of mass.



Question for today:

How can we take advantage of this intuition in the worst-case model?



- Suppose we received y , and we know there was a p -fraction of adversarial errors.
- Then y may have originated from any codeword in the shaded circle: either c or c' .
- If we have the intuition (from before) that "most of the mass is in the in-between spaces" then there should not be that many codewords in the shaded circle: mostly it just captures empty space.

This discussion motivates LIST DECODING.

We may not know which of c, c' was the right answer, but at least we have a pretty short list.

① LIST DECODING

DEF. A code $\mathcal{C} \subseteq \Sigma^n$ is (p, L) -LIST-DECODABLE if $\forall y \in \Sigma^n$,
 $|\{c \in \mathcal{C} : \delta(c, y) \leq p\}| \leq L$.

So if \mathcal{C} is (p, L) -list-decodable and there are a p -fraction of adversarial errors, we can narrow down the possibilities to L possible messages.

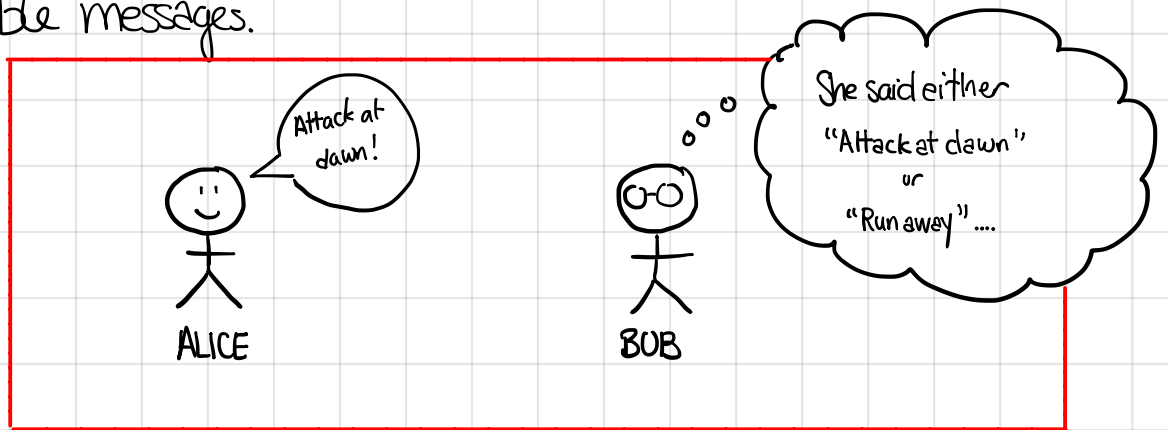


Fig 1. Not the most compelling application of list decoding.

Why might this be a good thing?

- In communication, if Bob can get some side information and/or use some crypto assumptions, he can narrow the list down.
- We see many other applications later.



Nonetheless, this is obviously only interesting in L is small.

So the question is:

WHAT IS THE BEST TRADE-OFF BETWEEN R, p, L ?

② LIST-DECODING CAPACITY THEOREM

THM (List-decoding Capacity) Let $q \geq 2$, $0 \leq p \leq 1 - 1/q$, $\epsilon > 0$. Then:

(1) If $R \leq 1 - H_q(p) - \epsilon$, there exists a family of q -ary codes that are $(p, O(1/\epsilon))$ -List-Decodable.

(2) If $R > 1 - H_q(p) + \epsilon$, then every (p, L) -list-decodable code of length n has $L \geq q^{-\Omega(n)}$.

This should look very familiar! Just like Shannon's thm for the BSC!

proof. (sketch)

(1) Use a random code! Let $\text{ENC}: \Sigma^k \rightarrow \Sigma^n$ be completely random.

Fix $\Lambda \subseteq \Sigma^k$, $|\Lambda| = L+1$, and pick $y \in \Sigma^n$.

$$\mathbb{P}\{\text{ENC}(\Lambda) \subseteq B_q(y, p)\} = \left(\frac{\text{Vol}_q(pn, n)}{q^n} \right)^{L+1} \leq 2^{-n(1-H_q(p))(L+1)}$$

Now union bound:

$$\mathbb{P}\{\exists \Lambda, \exists y \text{ s.t. } \text{ENC}(\Lambda) \subseteq B_q(y, p)\} \leq \binom{q^k}{L+1} \cdot q^n \cdot q^{-n(1-H_q(p))(L+1)}$$

$$\leq q^{k(L+1) + n - n(1-H_q(p))(L+1)}$$

Choose $R = 1 - H_q(p) - \epsilon$

$$= q^n [R + 1 - H_q(p)] \cdot (L+1) + 1$$

$$= q^n (1 - \epsilon(L+1)) = q^{-\Omega(n)} \text{ if } L \approx 1/\epsilon.$$

So then $\mathcal{C} = \text{Im}(\text{ENC})$ is (p, L) list decodable whp.

pf. ctd.

(2) Suppose we have a code \mathcal{C} that has rate $R > 1 - H_q(p) + \epsilon$.
We need to show $\exists y$ s.t. $|\mathcal{C} \cap B_q(p, y)|$ is large.

IDEA: Pick a random y .

For a fixed $c \in \mathcal{C}$, we have

$$\mathbb{P}\{c \in B_q(p, y)\} \geq \frac{\text{Vol}_q(p, n)}{q^n} \approx q^{-n(1-H_q(p))}$$

So the expected number of codewords in a ball is

$$\begin{aligned} \mathbb{E}_y |\mathcal{C} \cap B(p, y)| &= \sum_{c \in \mathcal{C}} \mathbb{E} \mathbb{1}\{c \in B(p, y)\} \\ &\geq |\mathcal{C}| \cdot q^{-n(1-H_q(p))} \\ &= q^{k-n(1-H_q(p))} \\ &\stackrel{\text{assm. on } R}{\geq} q^{n(1-H_q(p)+\epsilon)-n(1-H_q(p))} \\ &= q^{\epsilon n} \end{aligned}$$

which is what we claimed.

Thus, LIST-DECODING gives us a worst-case way to achieve $R = 1 - H_q(p)$!

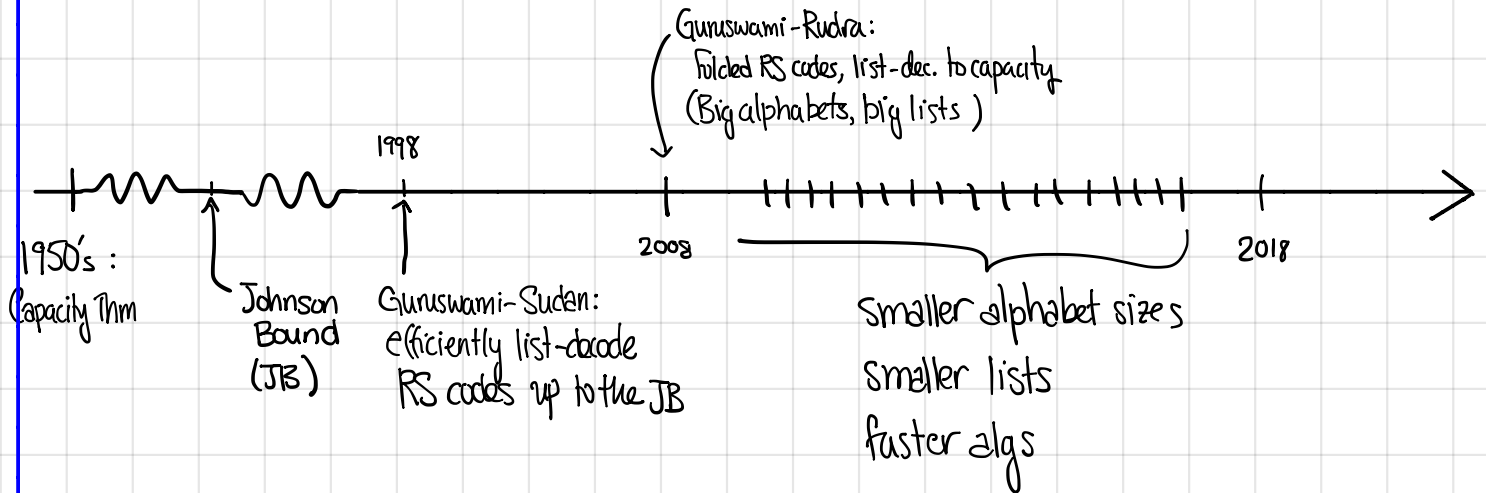
But as usual we have some questions.

1. Efficient Algorithms?
2. Explicit Constructions?
3. Small alphabet sizes?

ASIDE:

ALGORITHMIC LIST-DECODING

There's been lots of progress, but still there are many open questions.



State-of-the-art [as of Winter 2018]:

- Explicit constructions over constant-sized alphabets and ALMOST constant list sizes lists, with near-linear time algs. [Hemenway, RonZewi, W'17] ↖ There is a $\log^*(n)$ in it.
- (or, \exists Monte Carlo constructions w/ constant list size [Guruswami-Xing '12, '13])

Still Open:

- The "correct" constant-sized lists ($1/\epsilon$)
- Binary codes — we don't even have explicit constructions!

/end(ASIDE)

Let's start trying to answer Questions 1 and 2.

First try:

We have codes with good distance!
Isn't that enough?

③ JOHNSON BOUND

Suppose we have a code with good pairwise distance. That should say SOMETHING about list-decoding, right?

THM (JOHNSON BOUND)

$$\text{Let } J_q(\delta) = (1 - 1/q) \left(1 - \sqrt{1 - q\delta/q-1} \right)$$

Let $\mathcal{C} \subseteq \Sigma^n$ ($w/ |E| = q$) be a code with relative distance δ .

If $p < J_q(\delta)$, then \mathcal{C} is $(p, q \cdot \delta \cdot n^2)$ -LIST-DECODABLE.

There are many different versions of the Johnson bound.

You'll prove one on your homework

For a few more, check out "EXTENSIONS to the JOHNSON BOUND" (Guruswami, Sudan, 2001) which is posted on the website.

In class, let's just try to understand the statement. That $J_q(\delta)$ term is GROSS!

Let's start with $q=2$. How does the JB compare to capacity?

LIST-DECODING CAPACITY THM

If $R < 1 - H_2(p) - \epsilon$, then a random binary code of rate R is (p, L) -list-decodable for reasonable L .

vs.

JOHNSON BOUND

If $p < J_2(\delta) = \frac{1}{2}(1 - \sqrt{1 - 2\delta})$ then any code of distance δ is (p, L) -list decodable for reasonable L .

In order to compare these we need some way to compare R and δ .
 Since this is a positive result (\exists a code st...), let's use the GV bound.

So for any δ , we know there \exists a code of rate $R = 1 - H_2(\delta)$ and dist. δ .
 With this, we have: aka $\delta = H_2^{-1}(1-R)$

LIST-DECODING CAPACITY THM

If $R < 1 - H_2(p) - \epsilon$, then a random binary code of rate R is (p, L) -list-decodable for reasonable L

vs.

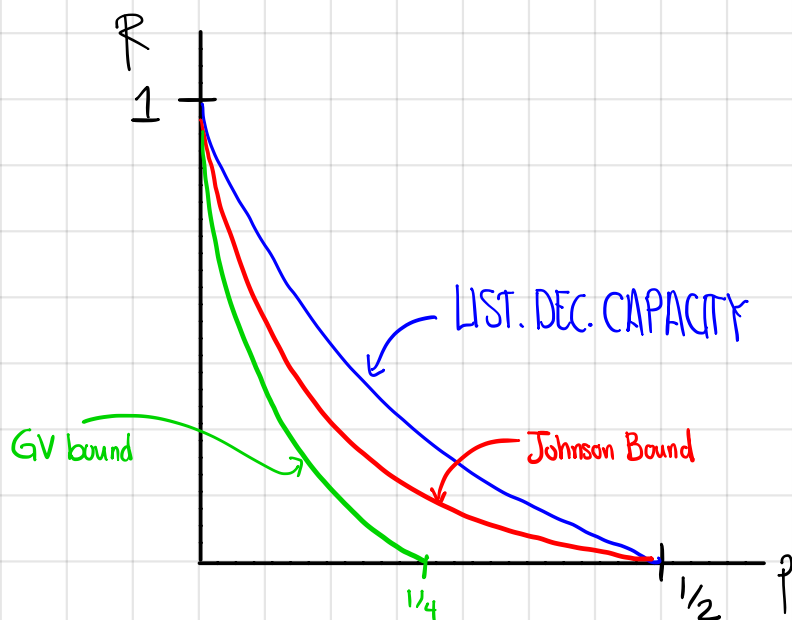
JOHNSON BOUND

If $p < J_2(\delta) = \frac{1}{2}(1 - \sqrt{1 - 2\delta})$ then any code of distance δ is (p, L) -list decodable for reasonable L .

If $p < \frac{1}{2}(1 - \sqrt{1 - 2H_2^{-1}(1-R)})$ then there exists a code of rate R that is (p, L) -list-decodable.

Solving for p gives: $R < 1 - H_2(2p(1-p))$

We can plot these two trade-offs:



So the Johnson Bound is WORSE than the List-decoding Capacity Thm...
 BUT it does let us get $p \rightarrow 1/2$ with positive rate.

And now we can do the same exercise for large q .

When q is really big, $J_q(\delta) = (1 - 1/q) \left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right) \approx 1 - \sqrt{1 - \delta}$

Moreover, $1 - H_q(p) \approx 1 - p$.

Again, we need some way to convert δ to R so let's use the SINGLETON BOUND and set $R = 1 - \delta$ in the Johnson Bound.

LIST-DECODING CAPACITY THM

If $R < 1 - p^{\text{ish}}$, then a random q -ary code of rate R is (p, L) -list-decodable for reasonable L

vs.

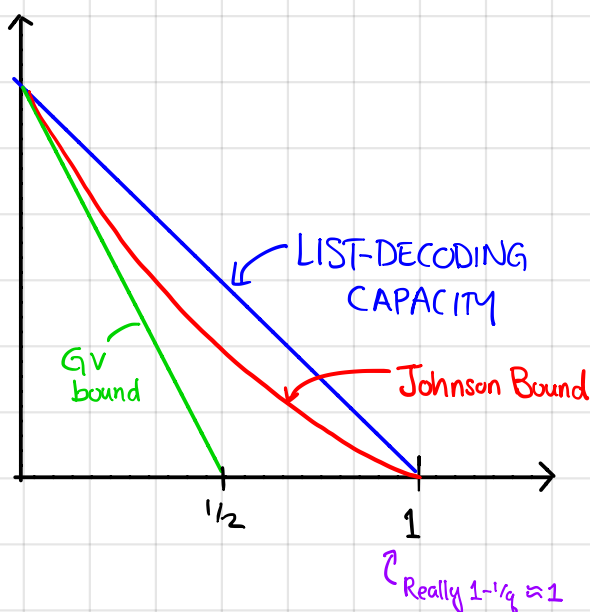
JOHNSON BOUND

If $p < J_2(\delta) \approx 1 - \sqrt{1 - \delta}$ then any code of distance δ is (p, L) -list decodable for reasonable L .



If $p < 1 - \sqrt{R}$ (aka, $R < (1 - p)^2$), there exists a code of rate R that is (p, L) -list-decodable for reasonable L .

Now, the picture looks like:



IN BOTH CASES ($q=2, q \rightarrow \infty$), the Johnson bound establishes that codes with good distance CAN be list decoded up to $1 - 1/q$ (instead of $\frac{1-1/q}{2}$, which is where unique decoding breaks).

However, the trade-off that we get isn't quite as good as list-decoding capacity.

QUESTIONS to PONDER

- ① What does the Johnson bound say about RS codes?
- ② Is it possible to prove ^{a non-vacuous statement of the form} that any code with good enough distance achieves list-decoding capacity?
- ③ Today we waved our hands about the connection between list-decoding and the Shannon model. Can you make this connection less hand-wavy?