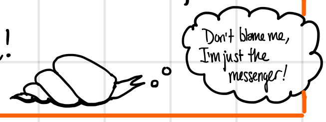
CS 250/EE 387 - LECTURE 8 - APPLICATIONS of CONCATENATED RS CODES

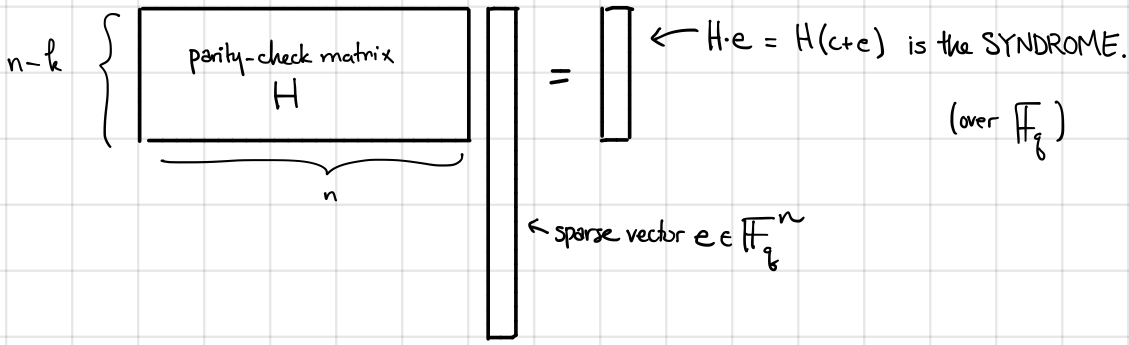
AGENDA

1. Syndrome decoding, sparse recovery, group testing
2. Application: group testing

GASTROPOD FACT.
Freshwater snails are popular hosts for many human and animal parasites – in particular, if you've ever gotten "swimmer's itch," a snail may be to blame!

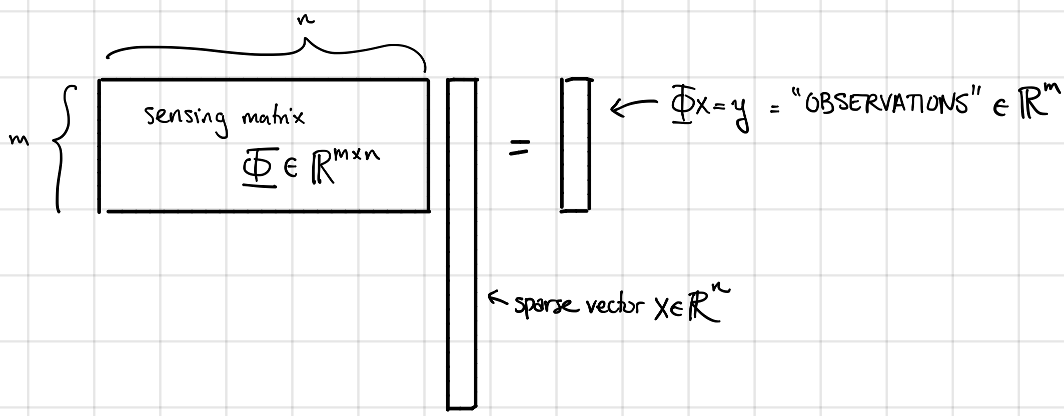


① The SYNDROME DECODING problem that we've seen a few times now is:



PROBLEM: Given $H \cdot e$, recover e.
GOAL: Make $n-k$ as small as possible.

This set-up might look familiar.
ONE SYNTACTICALLY SIMILAR PROBLEM: SPARSE RECOVERY / COMPRESSED SENSING.



PROBLEM: Given Φx, recover x.
GOAL: Make m as small as possible.

Why might we care about this?

1. Image processing and signal processing.

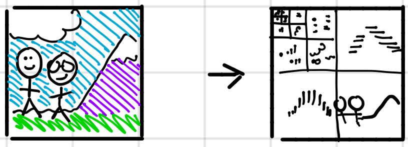


Image (not sparse) → Appropriate change-of-basis

- Most natural images/signals are sparse(ish) in some basis (or w/r/t some dictionary).
- So if we can acquire that image/signal by just measuring linear combinations of it and storing those, we can save time and space.

2. Streaming algorithms:

Consider a data stream:

$x_1, x_2, x_3, \dots, x_t, \dots \in$ some universe $\mathcal{U}$ of size n

You are interested in the frequency counts f_i = # times $i \in \mathcal{U}$ showed up.

But you don't want to store the vector $f \in \mathbb{R}^n$, especially if only a few items show up often.

Instead, keep a SKETCH $\quad \boxed{\Phi}\, f = \text{sketch}$.

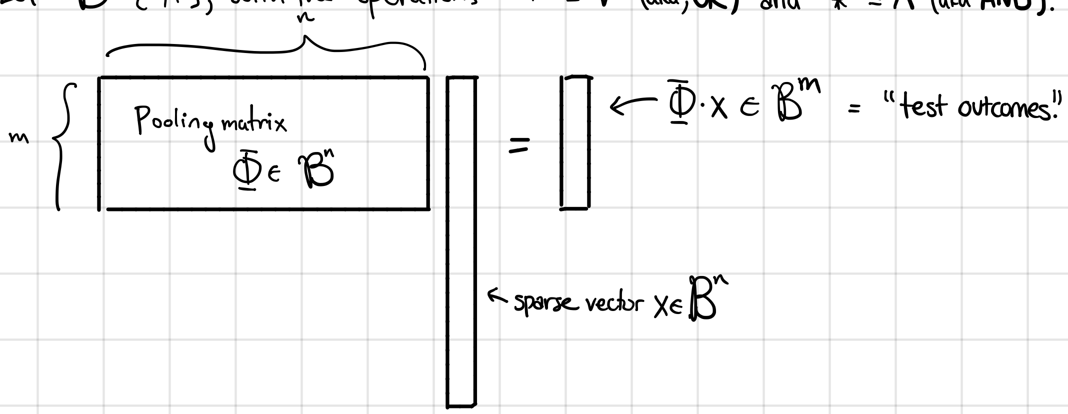
f is sparse(ish)

When a new item arrives, you can update the sketch by adding the appropriate column of Φ.

So this is exactly the same as syndrome decoding, except over $\mathbb{R}$ instead of $\mathbb{F}$.

ANOTHER SYNTACTICALLY SIMILAR PROBLEM: GROUP TESTING.

Let $\mathcal{B}=\{0,1\}$, with the operations "+" = $\vee$ (aka, OR) and "$*$" = $\wedge$ (aka AND).



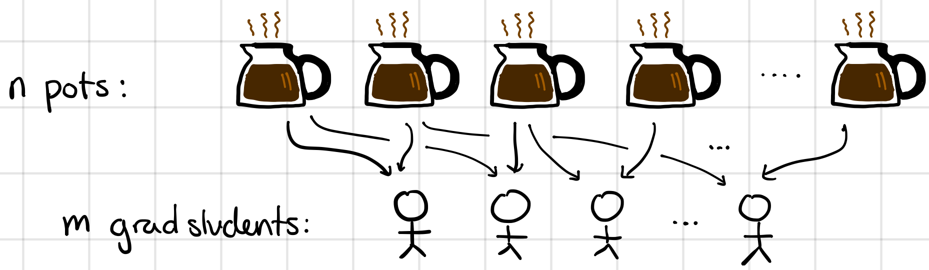
PROBLEM: Given $\Phi\cdot x$, recover x.

GOAL: make m as small as possible.

Why might we care about this?

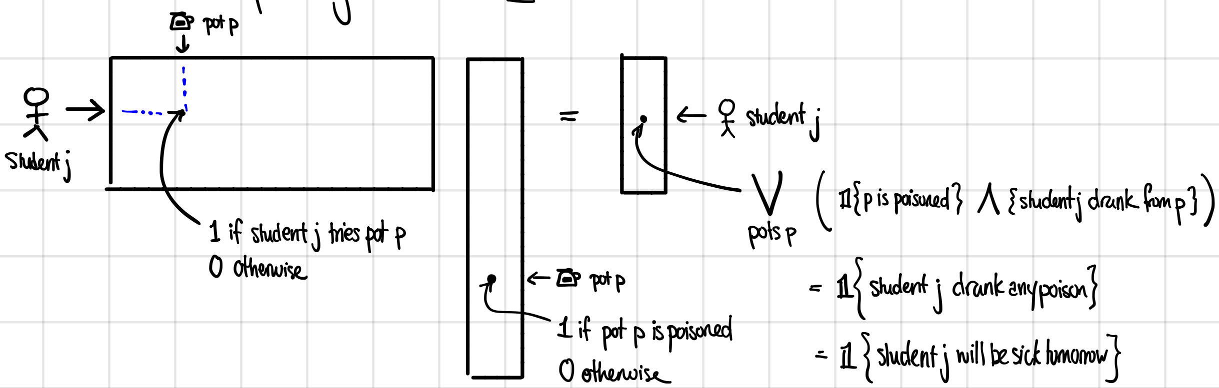
Suppose there are n pots of coffee:

- Unfortunately, $s \ll n$ of them are poisoned, but we don't know which.
- Fortunately, there are many graduate students available. If a grad student has even a drop of poisoned coffee today, then tomorrow they will be sick.
- You want to decide BY TOMORROW which pots of coffee are poisoned (so that you can drink the rest), while making as few grad students sick as possible (so as to maintain morale).
- The idea is to POOL the samples of coffee:



- If a student drinks from ANY poisoned coffee pot, they become sick.

Thus, if we make a "pooling matrix" Φ that is



So that's the picture we had before.

The PROBLEM is to recover x, the indicator vector of poisoned pots,
and the GOAL is to minimize the number of grad students subjected to this treatment.

MORE SERIOUSLY, this problem is usually motivated by biology.

- During WWII, the problem was introduced for testing US soldiers for syphilis.

soldiers $\longleftrightarrow$ coffee pots
blood sample $\longleftrightarrow$ coffee sample
syphilis tests $\longleftrightarrow$ graduate students

- Nowadays, for high-throughput screening.

civilians $\longleftrightarrow$ coffeepots
DNA samples $\longleftrightarrow$ coffee sample
genetic tests $\longleftrightarrow$ grad students

Tests are expensive, and not many soldiers/civilians are sick, so we'd like to use as few tests as possible.

So both GROUP TESTING and COMPRESSED SENSING are syntactically very similar to SYNDROME DECODING, it's just that they happen over $\mathbb{B}$, $\mathbb{R}$ or $\mathbb{C}$, and $\mathbb{F}_q$, respectively.

The different algebraic and geometric structures make these problems very different. However, ideas from one are often useful in others.

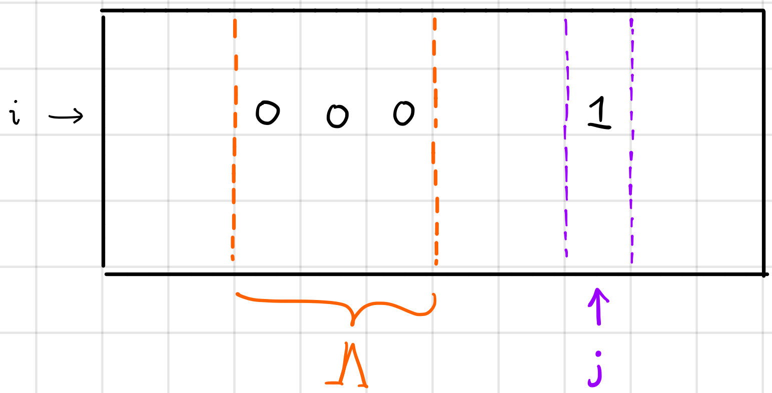
Today, we'll see how RS codes can be used to make good GROUP TESTING matrices.

DEF. A pooling matrix $\Phi \in \mathbb{B}^{m \times N}$ is d-disjunct if for any set $\Lambda \subseteq [N]$ of size d, and any $j \in [N] \setminus \Lambda$, there is at least one $i \in [m]$ so that:

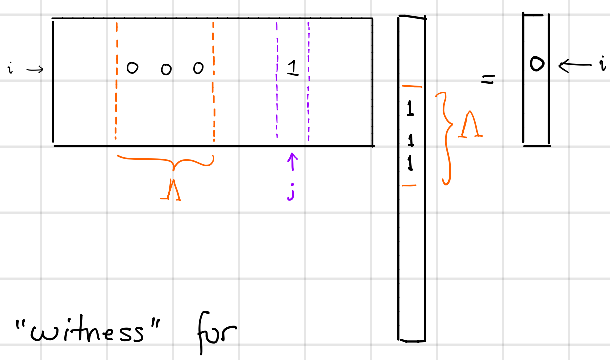
NOTE THE CHANGE to N. This will avoid notational collisions later.

$$\Phi_{ij} = 1 \quad \text{and} \quad \Phi_{i\ell} = 0 \quad \forall\, \ell \in \Lambda$$

Picture:



This is a good thing b/c if Λ we the true set of defectives (aka, poisoned coffeepots), Then



which gives a "witness" for j's status as not-poisoned.

THM. If Φ is d-disjunct, then as a pooling design it can identify d defectives.

Moreover, there is an algorithm that runs in time $O(m \cdot N)$ to identify the d defectives.

Pf. The algorithm is:

for each $j \in [N]$:
  if all the tests that j participates in are positive:
    label j as defective.
  else j is not defective.

Why does this work? Suppose that Λ is the true defective set and $j \notin \Lambda$. Then the def of d-disjunctness says that some test $i \in [m]$ which j participates in will come up negative, so the alg will label j "NOT DEFECTIVE."
OTOH, if $j \in \Lambda$, then by def. every test it participates in will be positive, so the alg will label j "DEFECTIVE."

So the goal is to come up with d-disjunct matrices $\Phi \in \mathcal{B}^{m \times N}$ so that m is as small as possible.

BEST CONSTRUCTIONS KNOWN: $m = O(d^2 \log_d(N))$ [Kautz-Singleton '64] - we'll see this today (based on RS codes)

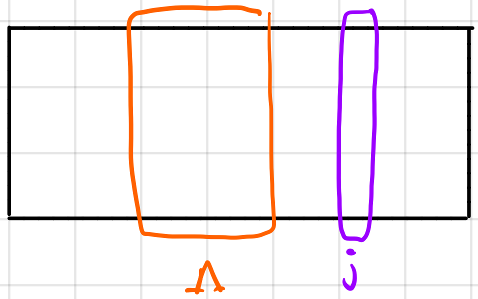
$m = O(d^2 \log(N))$ A random matrix does this - or check out [Porat-Rothschild '08] for an explicit construction. (also based on coding theory).

BEST LOWER BOUNDS: $m = \Omega\left(d^2 \log_d(N)\right)$ [Dyachkov-Rykov '86]

ALGORITHMS: If $m = O(d^2 \log(N))$, there's an EXPLICIT construction w/ SUBLINEAR TIME algorithm. [Ngo-Porat-Rudra '11(?)] (Also based on coding theory).
We'll see some faster algs later in the course.

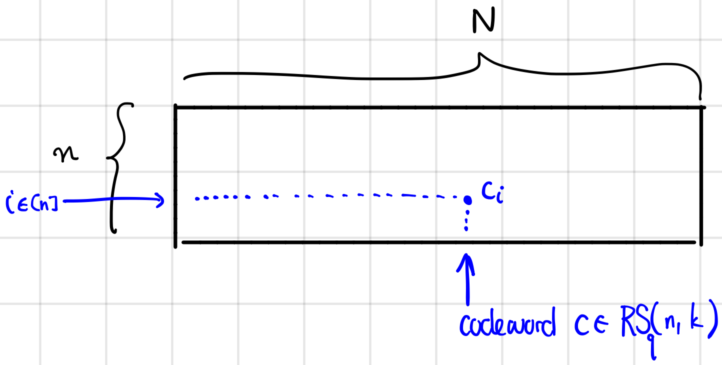
Today: A construction with $m = O\left(d^2 \log_d^2(N)\right)$.

IDEA:



We'd like all these columns to be kind of far apart... let's use codewords!

Let $\mathcal{C} = RS_q(n,k)$, let $N = q^k$, $m = q \cdot n$. Consider the matrix formed by:



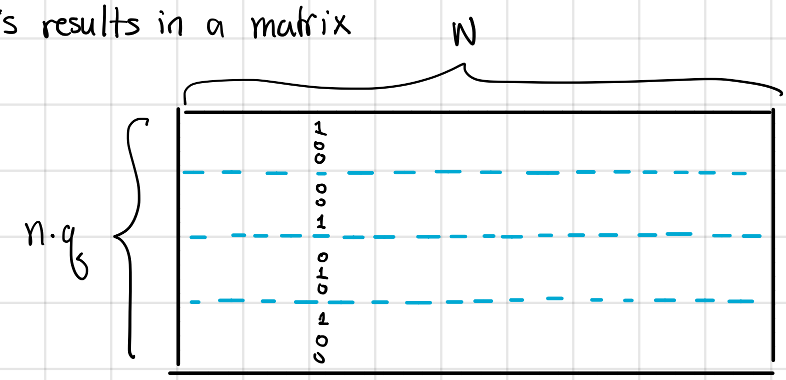
$\in \mathbb{F}_q^{n \times N}$

Now replace each symbol $\alpha \in \mathbb{F}_q$ w/ a vector of length q.

$$\alpha_1 \longleftrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \alpha_2 \longleftrightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \ldots, \quad \alpha_q \longleftrightarrow \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

where $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$.

This results in a matrix



aka, we've CONCATENATED $RS_q(n,k)$ with the code $\alpha_i \in \mathbb{F}_q \longmapsto e_i \in \mathbb{F}_2^q$

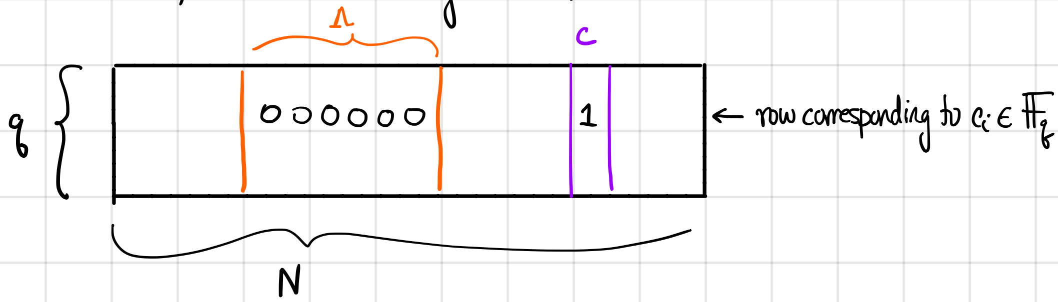
THM. If $\mathrm{dist}(\mathcal{C}) > n \cdot \left(\frac{d-1}{d}\right)$, then the matrix Φ obtained this way is d-disjunct.

Proof (by picture)

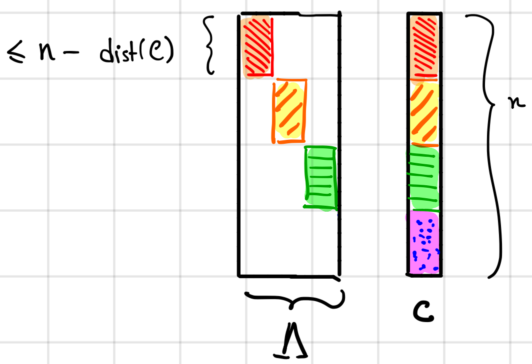
Because of how the construction works, we need to show:

$$\forall \quad \Lambda \subseteq \mathcal{C}, \quad |\Lambda| \le d, \quad \forall \quad c \in \mathcal{C} \setminus \Lambda, \quad \exists i \in [n] \text{ s.t. } c_i \notin \{ w_i : w \in \Lambda \}.$$

Indeed, if that were true, then the i^{th} layer would look like



So consider any $\Lambda \subseteq \mathcal{C}$, $|\Lambda| \le d$, and any $c \in \mathcal{C} \setminus \Lambda$.



- The first column of Λ agrees w/ c in at most $n - \mathrm{dist}(\mathcal{C})$ places: these ones in the picture
- The second column of Λ agrees w/ c (and not w/ the first col) in $\le n - \mathrm{dist}(\mathcal{C})$ places. these ones
- etc.

Altogether, there are at most $|\Lambda| \cdot (n - \mathrm{dist}(\mathcal{C}))$ positions of c that are agreed with by SOME column of Λ.

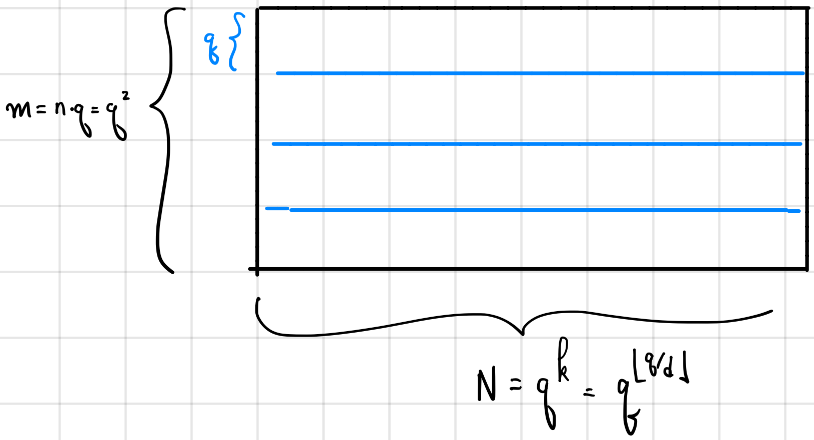
By our guarantee on $\mathrm{dist}(\mathcal{C})$, $\quad |\Lambda|(n - \mathrm{dist}(\mathcal{C})) < d\left(n - n\left(\frac{d-1}{d}\right)\right) = n$.

So there's at least one position that's not agreed with!

Let's instantiate this with $RS_q(\mathbb{F}_q, q, k)$, so $n = q$, $\mathrm{dist}(C) = q - k + 1$.

Setting $\mathrm{dist}(C) = n\left(\frac{d-1}{d}\right) + 1 = q\left(\frac{d-1}{d}\right) + 1$, we get $k = \lfloor q/d \rfloor$.

Then our matrix is:



Thus we choose $q = \sqrt{m}$, which implies $\log_q(N) = \left\lfloor \frac{\sqrt{m}}{d} \right\rfloor$ aka, $\sqrt{m} \approx d \log_q(N)$.

Then $m \approx d^2 \dfrac{\log^2(N)}{\frac{1}{2}\log^2(m)}$ which implies $m = O\left(d^2 \dfrac{\log(N)}{\log(d)}\right)$,

as claimed.

③ Quick Note About Compressed Sensing.

A very similar construction can be used to get deterministic compressed sensing matrices.

For those who know the lingo, this EXACT SAME construction appropriately normalized is an S-RIP matrix $\Phi \in \mathbb{R}^{m \times N}$ with $m = O(s^2 \log^2(N))$.

And you can do slightly better if you replace $\mathbb{F}_p$ w/ the p^{th} roots of unity.

[See Cheraghchi's "Coding-Theoretic Methods for Sparse Recovery" for lots more!]

QUESTIONS TO PONDER

1. Can you come up with a recovery scheme for this group testing matrix that runs in time $\mathrm{poly}(d\log(N))$ [in particular, sublinear in n?]

2. Can you make a group testing or compressed sensing scheme using the semantic similarity to syndrome decoding? (Rather than the scheme we saw, which used a different connection to coding theory)