CS250/EE387 - LECTURE 2 - Linear Codes and Finite Fields.

AGENDA

0. RECAP from LAST TIME
1. LINEAR ALGEBRA over $\{0,1\}$?
2. FINITE FIELDS
3. LINEAR CODES

TODAY'S OCTOPUS FACT:

The oldest known octopus fossil is from an animal that lived almost 300 million years ago!

0 Recall all this notation we had from last time:

n : block length
k : message length $(k \le n)$
d : distance $(d \le n)$
Σ : alphabet

A CODE is a subset $C \subseteq \Sigma^n$. Its elements are called CODEWORDS.
If $|C| = |\Sigma|^k$, the RATE of C is k/n.

QUESTION from last time:

What is the best trade-off between rate and distance?

(Still open!)

In particular, recall EXAMPLE 3 from last time:

$ENC: \{0,1\}^4 \to \{0,1\}^7$, given by:

$$ENC: (x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3, x_4, x_2+x_3+x_4 \bmod 2, x_1+x_3+x_4 \bmod 2, x_1+x_2+x_4 \bmod 2)$$

$C := \mathrm{Image}(ENC)$.

C is a binary code of length 7, message length 4, distance 3, rate $R = 4/7$.
We say it is a $(\underset{n}{7}, \underset{k}{4}, \underset{d}{3})_{\underset{|\Sigma|}{2}}$ code.

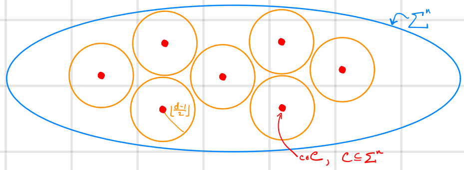
We called this the HAMMING CODE (of length 7) and we saw that it was optimal in that it met the HAMMING BOUND.

Recall

HAMMING BOUND

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

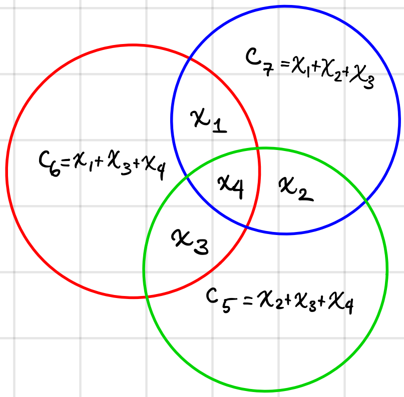
which followed from the fact that you can't pack too many Hamming balls in Σ^n:



We also came up (sort of) with a decoding algorithm for this code:

Algorithm Sketch:

View the code like this:



Then identify which circles don't sum to 0 (mod 2) and flip the unique bit that ameliorates the situation.

We waved our hands at how this sort of argument can also show that the distance is at least 3.

But this was all a bit unsatisfying. While clever, this construction feels a bit ad hoc.
How can we generalize this construction?
How can we generalize this algorithm / distance argument?

Today we'll see an important framework in coding theory, that of LINEAR CODES, which will help us put this example in context.

ASIDE So far we've mentioned the Hamming model, Hamming bound, Hamming distance, Hamming balls, and Hamming codes. Who was this guy Hamming?

Richard Hamming (1915 – 1998) was working at Bell labs starting in the late 1940's, where he was colleagues with Claude Shannon (of the "Shannon model" which we also mentioned).

Hamming was working on old-school computers (calculating machines), and they would return an error if even one bit was entered in error. This was extremely frustrating, and inspired Hamming to study this rate-vs-distance question, and to come up with Hamming codes.

① LINEAR ALGEBRA over $\{0,1\}$?

EXAMPLE 3 (from now on, THE HAMMING CODE) has a really nice form:

$$\mathrm{ENC}: (x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3, x_4, x_2+x_3+x_4, x_1+x_3+x_4, x_1+x_2+x_4)$$

mod 2

$$\mathrm{ENC}: \vec{x} \mapsto (\vec{x}, \text{ some linear fn of } x \pmod 2).$$

aka, we can write this as $x \mapsto Gx \pmod 2$, where G is some matrix.

$$\mathrm{ENC}(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}}_{G} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \pmod 2$$

G is called a GENERATOR MATRIX.

SUPPOSE FOR NOW that "linear algebra works mod 2." Then this view is pretty useful.

NOTE: Some people write

$$x \; \boxed{\;G\;} = c$$

aka, G is short and fat.

In this class, generator matrices are tall and skinny.

Why is this a useful way to look at things?

Let us pretend that linear algebra "works" mod 2, and see what we can do.

LINEAR CODES

$\mathcal{C}$ has a very nice property: it is closed under addition: if $c \in \mathcal{C}$, $c' \in \mathcal{C}$, then $c + c' \in \mathcal{C}$.
↑ mod 2 ↑ mod 2

This view makes that very clear:

$$\underbrace{\boxed{G}\,\boxed{}_{x}}_{c} + \underbrace{\boxed{G}\,\boxed{}_{x'}}_{c'} = \underbrace{\boxed{G}\,\boxed{}_{x+x'}}_{\text{also in } \mathcal{C}}$$

Aka, $\mathcal{C} = \text{span}(\text{cols}(G))$ is a LINEAR SUBSPACE, of DIMENSION 4.

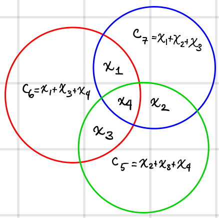
VERY USEFUL OBSERVATION

If $\mathcal{C}$ is LINEAR, then distance$(\mathcal{C}) = \min \text{wt}(\mathcal{C})$. Indeed, $\Delta(Gx, Gx') = \Delta(G(x-x'), 0)$
$= \text{wt}(G(x-x'))$.

↖ This can make it much easier to understand the distance.

PARITY CHECK MATRICES.

The other way we looked at this example was



We observed that all the circles summed to 0 mod 2.
Another way of writing that:

$$\begin{array}{r} \text{Green Circle's Constraint} \rightarrow \\ \text{Red Circle's Constraint} \rightarrow \\ \text{Blue Circle's Constraint} \rightarrow \end{array} \underbrace{\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{H} \overset{c}{\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix}} = \underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{0} \mod 2.$$

Aka, $c \in \mathcal{C} \Rightarrow Hc = 0.$ (mod 2)

Aka, $\mathcal{C} \subseteq \text{Ker}(H)$.

QUESTION: Does $\mathcal{C} = \text{Ker}(H)$?

ANSWER YES, $C = \mathrm{Ker}(H)$.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

It's easy to see $\dim(\mathrm{colspan}(G)) = 4$ since the identity matrix is just sitting there.

Why? Dimension counting! $\dim(C) = 4$.

- $\dim(\mathrm{Ker}(H)) = 7 - \dim(\mathrm{rowspan}(H)) = 7 - 3 = 4$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Again, it's easy to see $\dim(\mathrm{rowspan}(H)) = 3$ because of the identity.

- So $C \subseteq \mathrm{Ker}(H)$, and $\dim(C) = \dim(\mathrm{Ker}(H))$,
  $\Rightarrow C = \mathrm{Ker}(H)$

H is called a PARITY-CHECK matrix of C.

PARITY-CHECK MATRICES are USEFUL.

① It makes it easier to see the distance of C.

CLAIM: $\mathrm{dist}(C) = 3$

Proof:

As before, suffices to show $\min_{c \in C \setminus \{0\}} \mathrm{wt}(c) = 3$.

Suppose $c \in C$ has wt 1 or 2. Then

$$\underbrace{\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} \\ c \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$\mathrm{wt}(c) = 1$ or 2

But then either one column of H is 0 (NOPE)

or the sum of two columns of H is 0 mod 2

aka there is a repeated column (NOPE)

So $\forall c \in C$, $\mathrm{wt}(c) \geq 3$.

Now, the codeword 0101010 has weight exactly 3, so this is tight ☺

② It gives us a nice decoding algorithm.

Same puzzle as last time.

PUZZLE: Given $\tilde{c} = 0111010$ which has suffered one bit flip, what is c?

Solution to PUZZLE: Write $\tilde{c} = c + z \mod 2$ ← ERROR vector which has wt 1.

ON THE ONE HAND:

$$\underbrace{\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\tilde{c}} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \mod 2$$

ON THE OTHER HAND:

$$\begin{bmatrix} H \end{bmatrix} \left(\underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{c} + \underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{z} \right) = \begin{bmatrix} H \end{bmatrix} \underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{z} = \text{The column of } H \text{ where } z \text{ has a } 1.$$

Hz is called the "syndrome."

Since $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ is the 3^{rd} column of H, the error occured in position 3.

So this gives us an efficient decoding alg for $\mathcal{C}$!

This is a much nicer way of seeing our circle-based algorithm.
"Which circles fail to sum to 1" is the same as "which bits of $H(x+z)$ are 1", and it picks out which bit we needed to flip.

THE POINT SO FAR:

Assuming that "linear algebra works" in $\{0,1\}$ mod 2, this linear-algebraic view of things is very useful!

THE QUESTION: Does linear algebra "make sense" over $\{0,1\}$ mod 2?

(And what does that mean?)

What's the problem? Why wouldn't it work?
To see the (potential) issue, consider what happens for $\{0,1,2,3\}$ mod 4.

NON-EXAMPLE (WARNING! FALSE STATEMENTS BELOW)

Let $G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}$ be a generator matrix, mod 4.

Define. $\mathcal{C} = \left\{ \underset{\text{mod } 4}{G \cdot x} \,\middle|\, x \in \{0,1,2,3\}^2 \right\} = \text{colspan}(G)$.

So $\dim(\mathcal{C}) = 2$. (The columns are not scalar multiples of each other, aka, they are linearly independent)

But consider

$H = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$. Now we have $\underset{H}{\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}} \underset{G}{\begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ (mod 4)

And H certainly seems to have rank 2 also.
(The rows are not scalar multiples of each other).

So then by the same argument, $\mathcal{C} = \text{colspan}(G) = \text{Ker}(H)$.

So $2 = \dim(\mathcal{C}) = \dim(\text{Ker}(H)) = 3 - \dim(\text{rowspan}(H)) = 3 - 2 = 1$.

OH NO!!

WHY WAS THIS A NON-EXAMPLE?

What went wrong? Linear algebra does not "work" over $\{0,1,2,3\}$ mod 4.

ASIDE: You can make it work a little bit. The algebra buzzword is "module".

- In particular, several times in that example we said (something like):

"nonzero vectors v and w are linearly independent iff there is no λ s.t. $v = \lambda w$.

- Another definition of linear independence:

"nonzero vectors v and w are linearly independent iff there is no nonzero λ_1, λ_2 s.t. $\lambda_1 v + \lambda_2 w = 0$.

- Over $\mathbb{R}$, these are the same:

Proof: Suppose $\exists\, \lambda_1, \lambda_2 \neq 0$ s.t. $\lambda_1 v + \lambda_2 w = 0$. Then $v = \left(\frac{-\lambda_2}{\lambda_1}\right) w$.
Conversely, if $\exists\, \lambda$ s.t. $v = \lambda w$, then choose $\lambda_2 = \lambda$, $\lambda_1 = -1$ and $\lambda_1 v + \lambda_2 w = 0$.

- But over $\{0,1,2,3\}$ mod 4, these are not the same.

$$2 \cdot \underbrace{\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}}_{v} + 2 \cdot \underbrace{\begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}}_{w} = \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \bmod 4,$$

even though v and w are not scalar multiples of each other.

- The Proof above breaks: What does $\left(\frac{-\lambda_2}{\lambda_1}\right)$ mean?

($3/2$ mod 4 does not immediately make sense).

This does not bode well for algebraic coding theory if even linear algebra doesn't work...

② FINITE FIELDS

FORTUNATELY, all that stuff that we did mod 2 actually was OK!

The difference between $\{0,1,2,3\}$ mod 4 and $\{0,1\}$ mod 2 is that $\{0,1\}$ mod 2 is a **FINITE FIELD.**

Informal definition of a field:

> A **FIELD** is any set of elements that you can add, subtract, multiply and divide like you want to.

Formal definition of a field:

DEF A FIELD $\mathbb{F}$ is a set of elements, along with operations $+$, $\times$, ("addition" and "multiplication") so that:

$\forall\ x,y,z \in \mathbb{F}$:

- (ASSOCIATIVITY) $(x+y)+z = x+(y+z)$
  $(x\times y)\times z = x\times(y\times z)$
- (COMMUTATIVITY) $x+y = y+x$. $x\times y = y\times x$.
- (DISTRIBUTIVITY) $x\times(y+z) = (x\times y)+(x\times z)$
- (IDENTITIES) There is an element "0" and an element "1" so that
  $x+0 = x \quad \forall x\in\mathbb{F}$
  $x\cdot 1 = x \quad \forall x\in\mathbb{F}$
- (INVERSES) $\forall x\in\mathbb{F}$, $\exists y$ s.t. $x+y=0$ (Let's call this y "$-x$")
  $\forall x\in\mathbb{F}, x\neq 0$, $\exists y$ s.t. $x\cdot y = 1$ (Let's call this y "$\frac{1}{x}$" or "x^{-1}")

Familiar examples of fields: $\mathbb{R}$, $\mathbb{C}$.

A **FINITE FIELD** is a finite field. (aka, a field that is finite).

Familiar example: $\{0,1\}$ mod 2.

(The only thing to check is the inverses: $-0=0$, $-1=1$, $1^{-1}=1$. so we're good!)

Familiar non-example: $\{0,1,2,3\}$ mod 4.

(2 has no multiplicative inverse:

$0\cdot 2=0$
$1\cdot 2=2$
$2\cdot 2=4\equiv 0 \mod 4$
$3\cdot 2=6\equiv 2 \mod 4$

There's no way to get 1!)

"THEOREM:" Linear algebra "works" ENOUGH over finite fields.

There are some things that don't.

Most notably, orthogonality doesn't mean what you think it means.

The vector $\begin{pmatrix}1\\1\end{pmatrix}$ is orthogonal to itself over $(\{0,1\} \mod 2)$! WEIRD.

Before we go into more details, WHEN DO FINITE FIELDS EXIST?

ARE WE STUCK IN $\{0,1\}$ mod 2?

Theorem. For every prime power p^t, there is a unique* finite field with p^t elements. We call this field $\mathbb{F}_{p^t}$.

There are no other finite fields.

*Up to appropriate similarities

some people call it $GF(p^t)$.
GF stands for "Galois Field".
I might use this sometimes.

Proof. Exercise.

(Not really — I'll post some reading if you are interested, but if you are not you can take this Thm on faith.

EXAMPLE $\mathbb{F}_5 = \{0,1,2,3,4\} \bmod 5.$

Again, the only interesting part is the inverses:

$1 \cdot 1 = 1$
$2 \cdot 3 = 1 \quad (6 \bmod 5)$
$3 \cdot 2 = 1$
$4 \cdot 4 = 1 \quad (16 \bmod 5)$

So, for example, $\frac{1}{2} = 3 \bmod 5$

$0 + 0 = 0$
$1 + 4 = 0$
$2 + 3 = 0$
$3 + 2 = 0$
$4 + 1 = 0$

So, for example, $-1 = 4 \bmod 5$.

More generally, $\mathbb{F}_p = \{0, 1, \ldots, p-1\} \bmod p.$

EXAMPLE $\mathbb{F}_4$ is NOT $\{0,1,2,3\} \bmod 4.$
Instead, it is $\{0, 1, \gamma, \gamma^2\}$, with:

$+$	0	1	γ	γ^2
0	0	1	γ	γ^2
1	1	0	γ^2	γ
γ	γ	γ^2	0	1
γ^2	γ^2	γ	1	0

$\times$	0	1	γ	γ^2
0	0	0	0	0
1	0	1	γ	γ^2
γ	0	γ	γ^2	1
γ^2	0	γ^2	1	γ

FUN EXERCISE: Check that this satisfies all the axioms.

More generally, $\mathbb{F}_{p^t}$ is NOT the same as $\{0, 1, \ldots, p^t - 1\} \bmod p^t$ when $t > 1$.

FUN EXERCISE: If you haven't seen finite fields before, prove both of the "more generally" statements.

③ LINEAR CODES

Now that we have the appropriate language about finite fields, we can formally define the things we were talking about before with the Hamming code.

All the definitions you know + love for linear algebra over $\mathbb{R}$ make sense over finite fields:

Let $\mathbb{F}$ be a finite field. Then:

FUN EXERCISE: Check that $\mathbb{F}^n$ and any subspace $V \subseteq \mathbb{F}^n$ is a VECTOR SPACE over $\mathbb{F}$ (in the sense that they satisfy the axioms of a vector space that you know and love).

and/or just read on Wikipedia: Vector_space#Definition

- $\mathbb{F}^n := \{(x_1, \ldots, x_n) : x_i \in \mathbb{F}\}$.
- A **SUBSPACE** $V \subseteq \mathbb{F}^n$ (over $\mathbb{F}$) is a subset that is closed under addition & scalar multiplication.
  aka, $\forall v, w \in V$, $\forall \lambda \in \mathbb{F}$, $v + \lambda w \in V$.
- Vectors $v_1, \ldots, v_t \in \mathbb{F}^n$ are **LINEARLY INDEPENDENT** if $\forall \lambda_1, \ldots, \lambda_t \in \mathbb{F}$ that are not all 0, $\sum_i \lambda_i v_i \neq 0$.
- For $v_1, \ldots, v_t \in \mathbb{F}^n$, their **SPAN** is $\mathrm{span}(v_1, \ldots, v_t) = \{\sum_i \lambda_i v_i : \lambda_i \in \mathbb{F}\}$
- A **BASIS** for a subspace $V \subseteq \mathbb{F}^n$ is a collection of vectors $v_1, \ldots, v_t \in V$ s.t.
  - $v_1, \ldots, v_t$ are linearly independent
  - $V = \mathrm{span}(v_1, \ldots, v_t)$.
- The **DIMENSION** of a subspace V is the number of elements in any basis of V.

FUN EXERCISE: Prove that this is well-defined. (eg, all bases have the same size).

DEF. A **LINEAR CODE** $\mathcal{C}$ of length n and dimension k over a finite field $\mathbb{F}$ is a k-dimensional linear subspace of $\mathbb{F}^n$. (The alphabet of $\mathcal{C}$ is $\Sigma = \mathbb{F}$)

NOTE: We have overloaded k (message length & dimension).
In fact this makes sense. If $\mathcal{C}$ is a k-dimensional subspace over $\mathbb{F}$, then $|\mathcal{C}| = |\mathbb{F}|^k$, hence $k = \log_{|\mathbb{F}|} |\mathcal{C}| = \log_{|\Sigma|} |\mathcal{C}|$ = message length.

Why? Every $c \in \mathcal{C}$ has a unique representation as $\sum_{i=1}^{k} \lambda_i v_i$ for a basis $v_1, \ldots, v_k$.
That's $|\mathbb{F}|^k$ choices for the λ_i.

OBSERVATION. If C is a linear code over $\mathbb{F}$, then there is a matrix $G \in \mathbb{F}^{n \times k}$ so that $C = \{G \cdot x : x \in \mathbb{F}^k\} =: \text{colspan}(G)$.

pronounced "column span." The span of the columns of G

Proof of OBSERVATION: Choose the columns of G to be a basis of C.

DEF. A matrix $G \in \mathbb{F}^{n \times k}$ so that $C = \{G \cdot x : x \in \mathbb{F}^k\}$ is called a GENERATOR MATRIX for C.

Note: There can be many generator matrices for the same code. They all describe the same code, but they implicitly describe different encoding maps. For example,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

are both generator matrices for the Hamming code.

(FUN EXERCISE: Check!)

However, some generator matrices may be more useful than others. For example, G above corresponds to a SYSTEMATIC encoding map. This means that $\text{Enc}_G : (x_1, x_2, x_3, x_4) \mapsto (\underbrace{x_1, x_2, x_3, x_4}, \text{STUFF})$.

The message shows up as the first part of the codeword.

G' still corresponds to a legit encoding map, but it's not systematic.

DEF. If $C \subseteq \mathbb{F}^n$ is a linear code over $\mathbb{F}$, then $C^\perp$ is the DUAL CODE:

$$C^\perp = \{ v \in \mathbb{F}^n : \langle v, c \rangle = 0 \ \forall \ c \in C \}.$$

This is the standard inner product: $\langle v, c \rangle = \sum_{i=1}^n v_i \cdot c_i$

NOTE: If $\dim(C) = k$, then $\dim(C^\perp) = n - k$. (Just like over $\mathbb{R}$).

FUN EXERCISE: Prove this. (Given that bases and dimensions make sense)

OBSERVATION. If C is a linear code of dimension k over $\mathbb{F}$, then there is a matrix $H \in \mathbb{F}^{n-k \times n}$ so that

$$\mathcal{C} = \{ c \in \mathbb{F}^n : Hc = 0 \} \quad \text{aka } C = \mathrm{Ker}(H).$$

PROOF of OBSERVATION: Let H be a matrix whose rows are a basis for $C^\perp$.

DEF. A matrix $H \in \mathbb{F}^{(n-k) \times n}$ so that $\mathcal{C} = \{ c \in \mathbb{F}^n : H \cdot c = 0 \}$ is called a <u>PARITY CHECK</u> matrix for C.

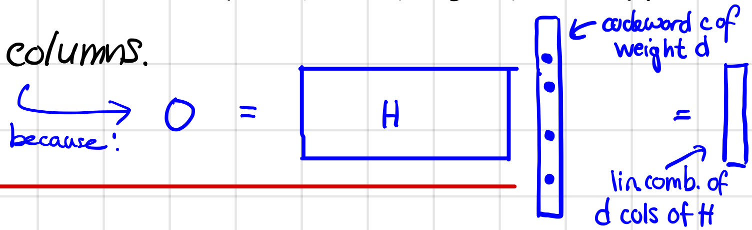
The rows of H (or any vector v s.t. $\langle v, c \rangle = 0 \ \forall c \in \mathcal{C}$) are called <u>PARITY CHECKS</u>.

<u>NOTE</u>: Again, there is not a unique parity check matrix for a code $\mathcal{C}$.

<u>SOME FACTS</u>: (FUN EXERCISE: Verify these!)

If $C \subseteq \mathbb{F}^n$ is a linear code over $\mathbb{F}$ of dimension k w/ generator matrix G and parity-check matrix H, then:

- $H \cdot G = 0$
- $C^\perp$ is a linear code of dimension $n-k$ with generator matrix H^T and parity-check matrix G^T.
- The distance of C is the minimum weight of any nonzero codeword in $\mathcal{C}$: $\mathrm{dist}(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \sum_{i=1}^n \mathbb{1}\{c_i \neq 0\}$.
- The distance of C is the smallest number d so that H has d linearly dependent columns.



That's all for today!

QUESTIONS TO PONDER

① Does there always exist a generator matrix G so that G looks like $\begin{array}{c} k \\ \left[\begin{array}{c} \ddots \\ \hline * \end{array}\right] \end{array} \begin{array}{l} \}k \\ \}n-k \end{array}$

If so, how would you find it efficiently?

What about nonlinear codes? Is there always an encoding map so that the message x appears as part of $\mathrm{ENC}(x)$?

Aka, the message x appears as the first part of $\mathrm{Enc}(x)$.

② How would you structure a linear code if you wanted to decode it <u>efficiently</u> from $\lfloor \frac{d-1}{2} \rfloor$ errors?

(What about generalizing the Hamming code that we saw?)