Lecture 10 - Error Correcting Codes

[Scribe] + refs [MacW-S, van Lint, Roth, Guru/Sudan/Rudra]

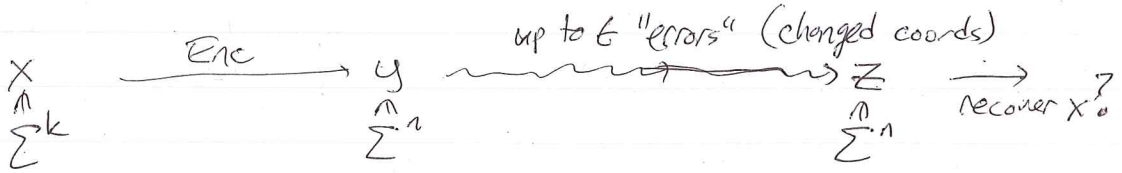
[Goal of ECCs: add small, clever amt of "redundancy" to data so that when stored/transmitted, if some of the bits/bytes corrupted, can still recover the orig. msg. Info thy / ECE people focus on rand. errs, TCS people (us) focus on worst-case errors. ||

def: An E.C.C. is an injective map $Enc: \Sigma^k \to \Sigma^n$

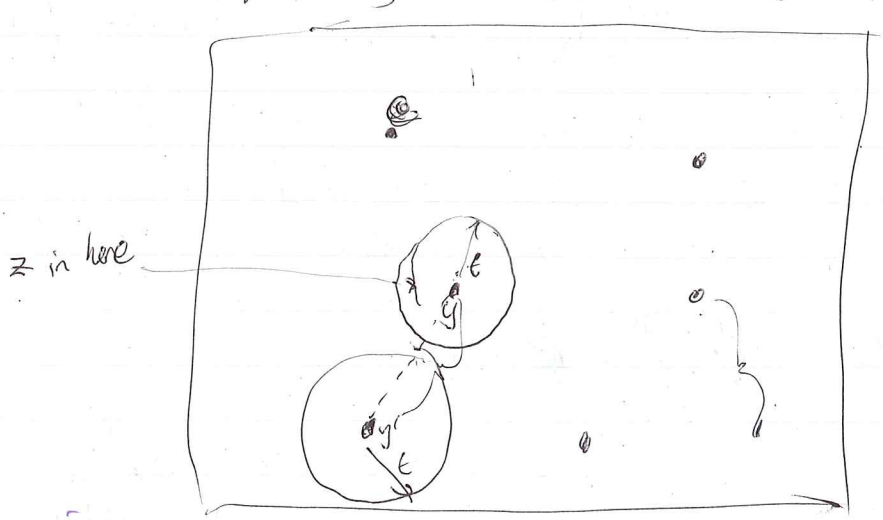
- Σ = alphabet, $q := |\Sigma|$, often 2
- Σ^k = message space. k = dimension / msg len.
- n = (block) length. $C := range(Enc) \subseteq \Sigma^n$ sometimes called the code. $y \in C$ a codeword
- $\frac{k}{n}$ = rate. Want it large [for efficiency's sake] ||

[don't insist on "systematic" codes]

[Fine, so why not just make $n=k$? Want to handle errors. ||



def: $\Delta(y,z)$ [Hamming distance] $= \#\{i : y_i \neq z_i\}$



$y \in C$

[Higher rate = more packed in ||

$\min_{y \neq y' \in C} \{\Delta(y,y')\}$

def: min distance $= d := \min_{x \neq x'} \{\Delta(Enc(x), Enc(x'))\}$ ← want large; [tension with rate]

fact: "Unique decoding" possible iff $t \leq \lfloor \frac{d}{2} \rfloor$

Q: Why not choose $C \subseteq \mathbb{F}_q^n$ of size q^k randomly? [It will give a great dist/rate tradeoff?]

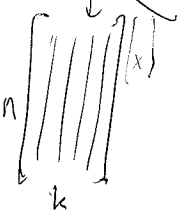
A: Lacks efficient [poly time] encoding (and decoding). [We want "explicit" codes w/ good tradeoff.]

[All the best codes we know are "linear codes".]

def: Linear code: $Enc: \mathbb{F}_q^k \to \mathbb{F}_q^n$, [hence $k = \dim$]

$x \mapsto Gx$ ← full rank

"generator mtx"



$C = Im(G)$ is k-dim subspace of $\mathbb{F}_q^n$

[rem: columns are k codewords, code is their span]

rem: Given G, encoding is efficient. [Yay, just mtx multiply.]

Notation: a $[n, k, d]_q$ code [if not linear, $[\cdot]$ becomes $(\cdot)$.]

[Decoding: in general NP-hard, but sometimes in P for nice codes]

[Checking for corruption is easy...]

[Another characterization of a subspace is all the vectors you're orthog to]

Let $C^\perp = \{w \in \mathbb{F}_q^n : w^T y = 0 \}$ ← way [dot prod.] Ex: subspace of dim $n-k$. (Thm 6.2.1) [$C^{\perp\perp} = C$]

↳ [gives rise to] $[n, n-k]_q$ code, $Enc^\perp: \mathbb{F}_q^{n-k} \to \mathbb{F}_q^n$, $w \mapsto H^T w$ "dual code"

where H is $(n-k)\times n$, the <u>parity check mtx</u> of Enc/C.

$C^\perp$ = rowspan(H) [= all vecs $\perp$ to everything in C]

$z \in C \iff Hz = \begin{bmatrix}0\\ \vdots \\ 0\end{bmatrix}\}n-k$

· "parity checks" if $q=2$.
· efficient check for "$\in C$?"

Linear code facts: · $d(C)$ = min. Hamming wt of nonzero codeword.

pf: $\Delta(y,y') = \text{HamWt}(y-y')$ ← a codeword when y, y' are! $\iff y=y'$.

= min. # cols of H which are linearly dependent

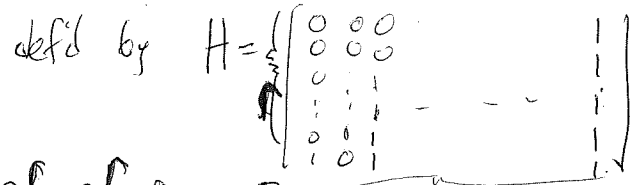
pf: $= \min\{ wt(z) : z \in C, z \neq 0 \}$.

$= \min\{ wt(z) : Hz = \vec{0}, z \neq 0 \}$.

[lin comb. of H's cols]

$q=2$ [binary codes?]

[As long as H doesn't have all-0 col, has all 1 cols (i.i.n.dep.
" " " " " " " two ident. cols," " 2 " " .
And for good rate we want H .. <u>fat</u> as poss.
So why not just take <u>all</u> distinct cols?]

def: <u>Hamming Code</u>: defd by $H =$  [all nonzero strings in lex order]

So it's a $[2^r-1,\ 2^r-r-1,\ 3]_2$ code. 2^r-1

n ↑ $k = n-\log_2(n)$

[full rank? = exercise] (contains all e_i's)

[23, 12, 7]$_2$ Golay code!

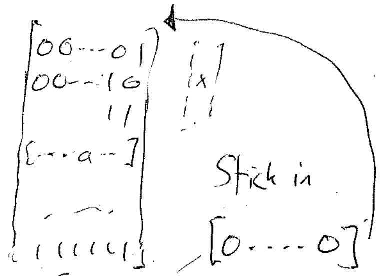
"Perfect code" [the rad-1 H. balls around each codeword partition]

Can unique-decode up to 1 error

$H(y+e_i) = Hy + (i \text{ written in binary})$ ← error

[[Hamming — awesome rate, pitiful distance.]] [[Reverse...??]]

<u>dual</u> of Hamming code has generator $H^T =$



[[kinda dumb, as codeword always starts w/ 0. Well, just do it, it's simpler]]

"Hadamard code"

$\Leftrightarrow$ def: Hadamard code: $x \longmapsto \left(\overset{a \cdot x}{x^T a} \right)_{a \in \mathbb{F}_2^r}$

def: For $x \in \mathbb{F}_2^r$, define $L_x : \mathbb{F}_2^r \to \mathbb{F}_2$,
$a \mapsto x^T a$

a linear polynomial, (deg. 1) $= \sum_{i=1}^{r} x_i a_i$ (coeffs, vbls!!)

Had: $x \longmapsto$ "truth table of L_x".

fact: $[2^r, r, \overset{2^{r-1}}{?}]_2$ code i.e., $[n, \log n, n/2]_2$

pf: Need all nonzero codewords have $\geq$ 1/2 coords nonzero.
I.e., $\forall x \neq 0$, $\Pr_{a \sim \mathbb{F}_2^r}[L_x(a) \neq 0] \geq \frac{1}{2}$.

✓ [[Schwartz-Zippel, or simply]] $\cdots x_1 a_1 + \cdots + x_i a_i + \cdots + x_r a_r$ (say nonzero. Choose all but a_i first ...)

ex: Generalize to $[q^r, r, (1-\frac{1}{q}) q^r]_q$.

[[Great dist! In fact, any binary code with dist $\geq (\frac{1}{2}+\varepsilon)n$ has $\leq O_\varepsilon(1)$ codewords, so optimal in a sense! Terrible rate!!]]

[Next up: Great rate, great distance! Bad.... alphabet size!]

def: Reed-Solomon Codes [Riv & Gus, mid-50s] [World's greatest, theory & practice ↝ DVDs, QR codes...]

For $1 \le k < n$, $q \ge n$, $S \subseteq \mathbb{F}_q$ with $|S| = n$...

$$\begin{aligned} \text{Enc}: \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ m &\longmapsto \left(P_m(a)\right)_{a \in S}, \text{ where } P_m(X) \in \mathbb{F}_q[X] \text{ is} \\ \| & \qquad m_0 + m_1 X + m_2 X^2 + \cdots + m_{k-1} X^{k-1} \\ (m_0, \ldots, m_{k-1}) & \end{aligned}$$

Rem: • usually $q = n$, $S = \mathbb{F}_q$, $m \longmapsto$ "truth table" of P_m.

• indeed linear [sum of two codewords is codeword]

• gen. mtx is "Vandermonde" [rows are $[\alpha^0, \alpha^1, \ldots, \alpha^{k-1}]$ as α varies in S]

• distance is ~~at least~~ $n - (k-1)$

Pf: min. Ham wt. of nonzero codeword $\ge n -$ max # 0's of nonzero codeword $\to$ Deg. Mantra!

$[n, k, n-k+1]_q$ (assuming $q \ge n$).

optimal

[proves Min dist. no bigger than $n-(k-1)$]

Singleton Bound: For $(n,k,d)_q$ code [not nec. linear], $k \le n - d + 1$

Pf: AFSOC $|C| > q^{n-d+1}$. PHP $\Rightarrow \exists\, y \neq y' \in C$ with same first $n-d+1$ coords.

Then $\Delta(y, y') \le d-1 < d$, $\Rightarrow\Leftarrow$. □

eg: $k = \frac{n}{2}$: rate $\frac{1}{2}$, (rel.) min dist $\approx \frac{1}{2}$ ☺ But $q \ge n$ ☹

(Asymptotically)
"Good codes". Fix q, let $\mathcal{C} = (C_n)$ be a seq. of $[n, k(n), d(n)]_q$ codes.

e.g. Hamming: $[n, n-\log(n+1), 3]_2$, $n = 2^r - 1$
Hadamard: $[n, \log n, n/2]_2$, $n = 2^r$
R.S.: $[n, k, n-k+1]_q$ (q not fixed ☹)

[asymptotic] rate, $R(\mathcal{C}) = \liminf_{n\to\infty} \frac{k(n)}{n}$

rel. distance, $\delta(\mathcal{C}) = \liminf_{n\to\infty} \frac{d(n)}{n}$.

e.g. $R(Ham) = 1$, $\delta(Ham) = 0$
$R(Had) = 0$, $\delta(Had) = \frac{1}{2}$.
$R(RS) = R_0$, $\delta(RS) = 1 - R_0$
("opt" as $q \to \infty$: Singleton)

def: "Asympt. Good Code fam." $R(\mathcal{C}) \geq R_0 > 0$, $\delta(\mathcal{C}) \geq \delta_0 > 0$. ($q$ fixed)

fact: They exist.
"Gilbert-Varshamov Bd": $\forall q$, $\forall \delta \in [0, 1 - \frac{1}{q}]$, $\exists \mathcal{C}$ ($\forall n$, infer) with $\delta(\mathcal{C}) = \delta$, $R(\mathcal{C}) \geq 1 - h_q(\delta) > 0$
- greedy, nonlinear constr.
- rand. linear code.

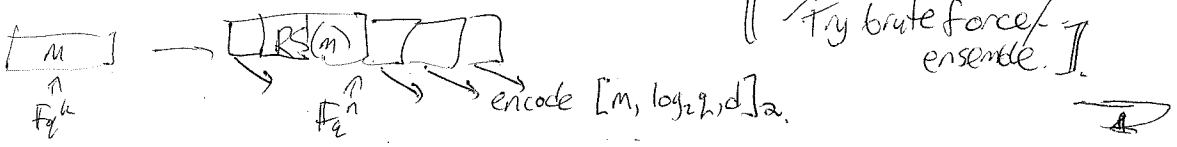
h_q "q-ary entropy"
[logs → give scribe formula $\delta \log_q \frac{1}{\delta} + (1-\delta)\log_q \frac{1}{1-\delta} + \delta \log_q (q-1)$]

Radius-δn Ham. ball has $\approx q^{h_q(\delta) n}$ pts.

[Rem: for $q \geq 49$ ∃ explicit codes BEATING GV based on alg. geom. !!!!]

thm: [Justesen '72] For $q = 2^r$, ∃ poly(n)-time constructible asympt. good codes ($\forall 0 < R_0 < 1$) ($\delta_0 \geq \frac{1}{2}(1 - R_0)$).
Also decodable from $\lfloor \frac{\delta_0}{2} n \rfloor$ errors in poly(n) time. [Venkat says!]

idea [Forney '66] "Concatenate" Reed-Solomon with small binary ("inner") code
[↗ Try brute force / ensemble.]



Concat: $[n_{outer}, k_{outer}, d_{outer}]_{q_{outer}} \diamond [n_{inner}, k_{inner}, d_{inner}]_{q_{inner}}$ → needs $q_{out} = q_{in}^{k_{in}}$

is $[n_{out} n_{in}, k_{out} k_{in}, \geq d_{out} d_{in}]_{q_{in}}$. ex: [R, S multiply]

[Sipser-Spielman '96]: Expander/Tanner codes:
binary, asymptotically good, can decode from some $\Omega(1)$ frac. of errors in $O(n)$ time!

RS $\diamond$ Hadamard: (concat.)

Had: $[q, \log q, \frac{1}{2} \cdot q]_2$ for $q = 2^r$

RS, $q := 2^r$, $[q, k, q-k+1]_q$.
[Yay there exists such a field!]

~~• bit message → q symbols → q^2~~

RS $\diamond$ Had: $[q^2, k \log q, \geq q(q-k+1)/2]_q$.

Write $n = q^2$ [4 integer], $k = \varepsilon q$:

$[n, \frac{1}{2}\varepsilon\sqrt{n}\log n, \geq \frac{1}{2}(1-\varepsilon)\cdot n]_2$.

$n \approx (\frac{k}{\varepsilon})^2$, k

Not bad! [can correct up to $\frac{n}{4}$ errors, encoding msgs of len k with $O(\frac{k}{\varepsilon})^2$ bits.]