ERROR-CORRECTING CODE

\[ C = \sum_{\Sigma}^{n} \sum_{\text{alphabet}}^{k, d} q - \text{code} \]

- block length \( n \)
- message length \( k \) (= \( \log_q |\Sigma| \))
- minimal distance \( d = \Delta(C) \)
- alphabet size \( q = \lvert \Sigma \rvert \)

Linear code: \( \Sigma = F_q \)
\( C \subseteq F_q^n \) linear subspace
Notation \( [n, k, d]_q \)

SINGLETEN BOUND: Any \( [n, k, d]_q \) - code has \( d \leq n-k+1 \)
Codes meeting this bound are maximum distance separable codes (MDS codes)

REED-SOLOMON CODE \( RS_q, n, k \)
\( q = \text{alphabet size} = \lvert F_q \rvert \)
\( n \leq q \) block length
\( k \leq n \) message length

- Pick \( n \) distinct elements \( x_1, x_2, \ldots, x_n \in F_q \)
- Identify message \( (c_0, c_1, \ldots, c_{k-1}) \in F_q^k \)
  with \( P_C = \sum_{j=0}^{k-1} c_j x^j \)
- Codeword is \( (P_C(x_1), P_C(x_2), \ldots, P_C(x_n)) \)

\( RS_q, n, k \) is an \( [n, k, n-k+1]_q \) - code (MDS)
REED-MULLER CODES \( RM_m, q \)

- alphabet size = \( |F_q| \)
- \( l \geq \) total degree of polynomials
- \( m \)-variate polynomials (over \( x_1, x_2, \ldots, x_m \))

**Case 1** \( l < q \)

- Message \( M(x) = \sum_{i_1, \ldots, i_m \leq l} m_{i_1, \ldots, i_m} x_1^{i_1} \ldots x_m^{i_m} \)

- Codeword \( (M(x))_{x \in F_q^m} \)

- Block length \( n = q^m \)

- Message length \( \binom{m + l}{m} \)

- Distance \( (1 - \frac{l}{q}) n \)

Fellows from

**SCHWARTZ-ZIPPERZ LEUUMA**

- Non-zero \( f \in F_q[x_1, \ldots, x_m] \), \( \deg(f) = d \)
- \( S \in F_q \), \( f \) is non-zero on at least \( 1 - \frac{d}{15} \) fraction of points in \( S^m \in F_q^m \)

**Case 2** \( l \geq q \)

- Messages polynomials of total degree \( l \)
- Individual degree \( \leq q - 1 \)

More complicated definitions and bounds

See notes from Lecture 9.
Today will talk about what it means that a code is "good"

Will see target parameters to shoot for in explicit constructions

Talk a little bit about how to compose codes to get new code

Finally discuss algorithmic challenge of efficient decoding (probably next 2 lectures)

But first... one more code you should know about (turns up in lots of different contexts)

HADAMARD CODES

Obtained from self-orthogonal matrices over \(\mathbb{F}_2\)

**Def.** An \(n \times n\) matrix \(H = \{h_{ij}\}\) is a HADAMARD MATRICE if \(h_{ij} \in \{+1, -1\}\) \(\forall i,j\) and \(HH^T = nI\) (in regular integer arithmetic)

Can view rows of Hadamard matrix as binary code of blocklength \(n\) with \(n\) codewords (i.e., message length \(\log n\))

Binary alphabet \(\{0, 1\} \leftrightarrow \{+1, -1\}\)

Distance? \(HH^T = nI\) means that for \(i \neq j\) \(\sum_{k=1}^n h_{ik}h_{jk} = 0\)

i.e. \(i\)th row and \(j\)th row equal in exactly half of pos distinct in exactly half of pos

So distance exactly \(n/2\)
can make code twice as large while keeping
distance by adding all complements

**Def 2.** Given an $n \times n$ Hadamard matrix $H$, the Hadamard Code of block length $n$, Had$n$, is the linear code whose codewords are the rows of $H$ (with $[1, -1]$ replaced by $[0, 1]$) and the complements of the rows of $H$.

**Prop 3.** For every $n \times n$ Hadamard matrix exists, the Hadamard code, Had$n$, is an $(n, \log (2n), n/2)_2$-code.

These parameters look somewhat similar to something achieved last time...

Namely, take Reed-Muller code RM$(m, 1, 2$)
- Total degree 1 (linear/affine functions)
- Over $F_2$
- gives a $[2^m, m+1, 2^{m-1}]_2$ code
(same parameters with $m = \log n$)

Is this a Hadamard code?

i.e., is there an underlying Hadamard matrix?

Yes!

The messages are coefficients $(c_0, c_1, \ldots, c_m)$ representing
$$ M(c) = c_0 + \sum_{i=1}^{m} c_i x_i $$

Look at codewords for $c_0 = 0$.
Will differ in exactly half the places $\leftrightarrow$ rows of Hadamard matrix.
Can view Hadamard code as all linear functions on $(x_1, x_2, ..., x_m)$.

[complement ⇔ affine functions]

Usual construction of Hadamard matrices for $n = 2^m$ is inductive

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_{m+1} = \begin{bmatrix} H_m & H_m \\ H_m & -H_m \end{bmatrix}$$

Summing up what codes we've seen so far

**HAMLING CODES**


**REED-SOLOMON CODES**

Optimal distance/message length behaviour. But require large alphabets. (Reed Muller decreases alphabet size a bit.)

**HADAMARD CODES**

Binary alphabet. Great distance! But very poor relationship between message length and block length.
Would like to find codes with
- constant ratio between message length & block length
- constant ratio between distance & block length

Need to study asymptotics of codes
Consider infinite families of codes
\[ C = \{ (n_i, k_i, d_i) \}, i \geq 1 \]
with \( \lim_{i \to \infty} \{ n_i \} = \infty \)
(and we'll want to think of \( q_i \) as fixed, usually)

**DEF 4** (MESSAGE RATE) \( R(C) = \liminf_{n \to \infty} \{ \frac{d_i}{n_i} \} \)

**DEF 5** A family of codes is ASYMMPTOTICALLY GOOD if \( R(C) > 0 \) and \( \delta(C) > 0 \).

Do asymptotically good codes exist? Yes.
Can they be constructed explicitly? Yes, and this was in fact achieved early on.
(Though to see one such construction before wrapping up our coding theory excursion)

Results in coding theory tend to have asymptotic versions interpretations.
Not seldom, these versions are more succinct.
Singleton bound \( d = n - k + 1 \)

**Singleton bound, asymptotic version**

\[ \delta = 1 - R \]

**Hamming bound (for binary codes)**

\[ 2^k \text{Vol}_2 \left( \frac{d-1}{2}, n \right) \leq 2^n \]

\[ \frac{d-1}{2} \times \frac{5n}{2} \]

\[ \text{Vol}_2 (pn, n) \times 2^k \cdot (p) \cdot n \]

for \( H(p) \) being the binary entropy function

\[ H(p) = -p \log_2 p - (1-p) \log_2 (1-p) \]

\[ = p \log (1/p) + (1-p) \log (1/(1-p)) \]

**Hamming bound, asymptotic version**

For \( q=2 \)

\[ R + H(\delta/2) \leq 1 \]

Random binary codes satisfy

\[ R \geq 1 - H(\delta) \]  \( (*) \)

(Doesn't meet Hamming bound)

There are no explicit constructions known achieving \( (*) \)
**Gilbert-Varshamov Bound**

Gilbert '52 : Random code satisfies \((\star)\)

Varshamov '57 : Random linear code satisfies \((\star)\)

**Gilbert:**

**Greedy** \((n, d)\)

\[ S := \{0, 1\}^n \]

\[ C := \emptyset \]

while \( S \neq \emptyset \)

Pick \( x \in S \)

\[ C := C \cup \{ x^2 \} \]

\[ S := S \setminus B(x, d) \]

**Prop 6** Fix \( \delta \in (0, 1/2) \) and \( \varepsilon > 0 \) and let \( R \geq 1 - H(\delta) - \varepsilon \). Then for all sufficiently large \( n \), GREEDY \((n, \lceil \delta n \rceil)\) produces a code with at least \( 2^{Rn} \) codewords.
Need that
\[ \text{Vol}_2 (p^n, n) = 2 \left( H(p) + o(1) \right) n \]

Proof of Prop 6.1: Pick \( n \) large enough so that
\[ \text{Vol}_2 (d,n) \leq 2 \left( H(d) + \varepsilon \right) n. \]
Assume the algorithm picks \( K \) codewords.
At every step, at most \( \text{Vol}_2 (d,n) \) elements removed from \( S \). Hence
\[ K \geq \frac{2^n}{\text{Vol}(d,n)} \geq 2 \left( 1 - H(d) - \varepsilon \right) n = 2 R_n. \]

Vazhekamov:

**Random-Linear \((n, k)\)**

Pick entries of \( G \in \mathbb{F}_2^{k \times n} \) uniformly
and independently at random.
Let \( C = \{ y \in \mathbb{F}_2^k | y \in \mathbb{F}_2^k \} \).

**Prop 7** Fix \( \delta \in (0, 1/2) \) and \( \varepsilon > 0 \) and let \( R = 1 - H(\delta) - \varepsilon \). Then for all sufficiently large \( n \) and \( k = \lceil Rn \rceil \) the procedure
**Random-Linear \((n, k)\)** produces a code with \( 2^k \) codewords and distance at least \( \delta n \) asymptotically almost surely.

**Proof** Need to prove two things:
1. \( G \) has full column rank \( k \) so that \( 2^k \) codewords
2. No codewords are closer than distance \( \delta n \)
Combine two claims into one:
For every $y \neq 0$, $yG \notin B(0, 5n)$

From this:
1. follows since $yG \neq 0$ for $y \neq 0$, so all rows linearly independent
2. follows since min distance of linear code
   = min weight of nonzero code word

Pick $n$ large enough so that
$$\text{Vol}_2(5n, n) \leq 2^{(H(\delta) + \epsilon/2)n}$$

Let $d = 5n$

For any fixed $y \neq 0$, $yG$ is random vector in $\mathbb{F}_2^n = \{0, 1\}^n$

Hence
$$\Pr[\text{wt}(yG) \leq d] = \Pr[yG \in B(0, d)]$$
$$= \frac{\text{Vol}_2(d, n)}{2^n}$$
$$\leq 2^{(H(\delta) + \epsilon/2 - 1)n}$$

Take union bound over all $y \in \mathbb{F}_2^k \setminus \{0\}$
$$\Pr[\exists y \neq 0 \text{ wt}(yG) \leq d] \leq 2^k 2^{(H(\delta) + \epsilon/2 - 1)n} \tag{\dagger}$$

If $R = k/n = 1 - H(\delta) - \epsilon$, then $(\dagger)$ becomes
$$\leq 2^{-\epsilon/2 \cdot n} \to 0 \text{ as } n \to \infty$$

and asymptotically almost surely $C$ has min distance $\geq 5n$.
How can one build asymptotically good codes? By combining two good codes to get an even better bound.

Outline from a few lectures ago
1. Build great code with too large alphabet
2. Smaller alphabet size while keeping overall goodness.

We saw a very simple example of this
1. Build Reed-Solomon code
2. Get down to binary alphabet by encoding binary strings as field elements

This increased block length but did not improve distance. Can we do something smarter?

Yes

Have code over large alphabet/field $\mathbb{F}_{q^k}$
Want to get down to alphabet $\mathbb{F}_q$

FACT: $\mathbb{F}_{q^k}$ can be viewed as vector space $\mathbb{F}_q^k$

There are linear bijections $\mathbb{F}_{q^k} \leftrightarrow \mathbb{F}_q^k$

Will use this heavily (but implicitly)
CONCATENATION (somewhat sloppy definition, but captures the essentials)

Have two codes

\[ C_1 \begin{bmatrix} n_1, k_1, d_1 \end{bmatrix}_{q_1} \]  
\[ C_2 \begin{bmatrix} n_2, k_2, d_2 \end{bmatrix}_q \]

outer code, large alphabet
inner code, small alphabet

CONCATENATION \( C_1 \circ C_2 \) is \( \begin{bmatrix} n_1, n_2, k_1, k_2, d_1, d_2 \end{bmatrix}_q \)-code
as defined next, using encoding functions

\[ E_1 : \mathbb{F}_{q_1}^{k_1} \to \mathbb{F}_{q_1}^{n_1} \]
\[ E_2 : \mathbb{F}_q^{k_2} \to \mathbb{F}_q^{n_2} \]

1. Take input \( x \) in \( \mathbb{F}_q^{k_2} \) (message)
2. View as element \( x' = (\mathbb{F}_q^{k_2})^{k_1} \) by linear injection
3. Encode by \( E_1 \) to outer codeword \( \beta \)
   in \( (\mathbb{F}_{q_1}^{k_1})^{n_1} \)
4. Interpret as \( n_1 \) messages \( (\mathbb{F}_q^{k_2})^{n_1} \) by linear injection
5. Encode coordinatewise using \( E_2 \)
   to get element \( (\mathbb{F}_q^{n_2})^{n_1} \)
6. The final codeword is this element \( y' \)
   in \( (\mathbb{F}_q^{n_2})^{n_1} = \mathbb{F}_q^{n_1n_2} \)
Prop 8. \( C_1 \otimes C_2 \) is an \( \text{is decodable} \) \([n_1, n_2, k, k_2, d_1, d_2]\) code.

Proof sketch:

Things to check:
- The code \( C_1 \otimes C_2 \) depends only on \( C_1, C_2 \) and not on \( E_1, E_2 \), or implicit linear bijections. We will completely ignore this.
- Message length and block length follows from construction.
- Linearity follows since all operators are linear.
- Distance: Need to show that non-zero codeword has weight \( \geq d_1, d_2 \)
  Suppose \( \alpha \in F_{k_1, k_2}^* \), \( \alpha \neq 0 \)
  Then \( \alpha' \in (F_{k_1, k_2})^k_1 \) is also \( \neq 0 \)
  \( C_1 \) has distance \( d_1 \), so \( \beta \) is non-zero in \( d_1 \) coordinates.
  \( \Rightarrow \beta' \) has \( d_1 \) non-zero messages

Every such message turns into a weight \( \geq d_2 \) codeword under \( E_2 \)
\( \Rightarrow \gamma \) has weight \( \geq d_1 d_2 \), QED.
Example 9 RS-Hadamard

Assume \( n = 2^m \)

Take \([n, k, n-k+1]_n\) Reed-Solomon code as outer code \( C_1 \)

Take \([n, \log n, n/2]_n\) Hadamard code as inner code \( C_2 \) (i.e., skipping complements for simplicity)

Obtain by concatenation

\[
[n^2, k \log n, \frac{n}{2} (n-k+1)]_2 - \text{code}
\]

Illustration

\( \epsilon \in \{0, 1\}^km \quad [m = \log n] \)

\( \epsilon \in \mathbb{F}_n^k \)

Encode with outer code

\( \epsilon \in \mathbb{F}_n^m \approx (\mathbb{F}_2^m)^n \)

Encode with inner code

\( \epsilon \in (\mathbb{F}_2^n)^n \)

\( n = 2^m \)

\( \mathbb{F}_n^{n^2} \)
Suppose we pick \( k = \Theta(n) \) in outer code.

Then \( C_1 \circ C_2 \) has

- constant relative distance
  \[
  \frac{n(n-k+1)/2}{n^2} \times 1 - \frac{k}{n}
  \]
- inverse polynomial rate
  \[
  \frac{\log n}{n^2} \times 1 \frac{\log n}{n}
  \]

New range of parameters compared to codes we've seen earlier.

What is the point of concatenation?
- Outer code can be over large alphabet
  \( \Rightarrow \) easier to construct
- Inner code allows us to shrink alphabet size

Can use multiple levels of concatenation to get better and better parameters

Won't get us all the way to asymptotically good codes, though.

Informally, for this we need both outer and inner codes to be "asymptotically good" in some sense.

Can use RS codes as outer codes, but need better inner codes. Details not hard given what we know now, but still beyond scope of our limited survey of coding theory.