Today:  
1. Linear Block codes (simplifies encoding/decoding tremendously!)  
2. Rectangular (Product) codes (Parity-check)

Last Time:  
Channel coding (ECC) — Two key ideas!

1. Embeddings:
   - place msgs geometrically in larger dimensional space to allow better distance between valid codewords

2. Parity Calculations:
   - Compute linear functions of Di's in msg!

(Alg.)

(Def.)  
1. Hamming Distance — btwn two codewords \( \vec{v} \) and \( \vec{w} \) is the number of coordinates in which they differ:
   - \( \text{HD}(\vec{v}, \vec{w}) = d_H(\vec{v}, \vec{w}) = |\{ i \mid v_i \neq w_i, i = 0, 1, \ldots, n-1 \}| \)
   - (Useful in determining code's error detection & correction capability)

\[ \begin{array}{c}
\vec{v} = 10101 \quad \text{HD}(\vec{v}, \vec{w}) = 2 \\
\vec{w} = 10011 \\
\end{array} \]

2. Weight \( w(\vec{v}) \) of a codeword is the number of non-zero coordinates in the codeword.
   - \( \vec{c} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \end{pmatrix} \)
   - \( w(\vec{c}) = \text{weight} \cdot 5 \)


3) Minimum Distance ($d_{\text{min}}$) of a block code

$\sim \min \text{ Hamming distance between all distinct pairs of codewords in } \mathcal{C}.$

$$d_{\text{min}} = \min \{ d_H(v_i^*, v_j^*) \} \times i \neq j.$$  

For an error to be undetectable, it must change the symbol values in at least $d_{\text{min}}$ coordinates for one codeword to look like another in $\mathcal{C};$

ie. Then: A code with $d_{\text{min}}$ can detect all error patterns of weight less than or equal to $(d_{\text{min}} - 1)$

$$t_{\text{detect}} \leq d_{\text{min}} - 1.$$  

(*) Consider, for example:

Want to send $\{\text{Apple (A), Banana (B)}\}$

Case I:

\[
\begin{align*}
\begin{array}{c}
A - 0 \\
B - 1
\end{array}
\end{align*}
\]

Distance $\Delta$ in codewords $= 1$

Thus cannot detect anything!

Case II:

\[
\begin{align*}
\begin{array}{c}
A - 00 \\
B - 11
\end{array}
\end{align*}
\]

Distance $\Delta$ in codewords $= 2$

Thus, can detect single error (e.g. 10 or 01 can detect, but cannot correct!)
Case III: $A = 000$  If Distance $d_{\min}$ codewords $= 3$

Thus can detect up to 2 errors
Can correct 1 error!

Continuing our observation,

it can be seen that

\[ t_{\text{detect}} \leq d_{\min} - 1 \] as before.

Also, then:

\[ t_{\text{correct}} \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \]

### Linear Block Codes

Recall: a $(n,k)$ block code of length $n$ contains $2^k$ codewords, where $k$ is the length of the original message.

Since there are $2^n$ possible $n$-bit words, there are $(2^n - 2^k)$ invalid words, i.e. $(2^n - 2^k)$ words not associated with codewords.

aka, the codebook contains redundancy!

Redundancy $r = n - k$

Code Rate $R = \frac{k}{n}$

Making code linear simplifies certain properties & makes implementation easier!
Properties of Linear Codes:

1. \( \tilde{c}_i + \tilde{c}_j \in \mathcal{C} \) (any linear combination of codewords is a codeword).
   - As a result, a **linear code always contains the all-zero vector**.

   **Example 1:** Is \( \mathcal{C} = \{(0000), (1111)\} \) linear? That is, is the length-4 binary repetition code linear?
   - Ans: Yes, any \( \tilde{c}_i + \tilde{c}_j \in \mathcal{C} \times \tilde{c}_i, \tilde{c}_j \).

2. Is \( \mathcal{C} = \{(00100), (10010), (01001), (11111)\} \) linear?
   - Ans: No, \( \tilde{c}_0 = (00000) \) not a codeword.

3. If \( \mathcal{C} \) is linear, then \( d_{\min} \) is the least weight non-zero codeword.
   - Pf: \( d_{\min} = \min \{ \| \tilde{c}_i \| \} \) and \( \| \tilde{c}_i - \tilde{c}_j \| = \min \{ \| \tilde{c}_i - \tilde{c}_j \| \} = \| \tilde{c}_k \| \) since code is linear, \( \tilde{c}_k \in \mathcal{C} \).

   **Example 2:** Consider \( \mathcal{C} = \{(00000), (00111), (01010), (01101)\} \)
   - Q1: Is this code linear? Ans: Yes, \( \tilde{c}_i + \tilde{c}_j \in \mathcal{C} \times \tilde{c}_i, \tilde{c}_j \).
   - Q2: What's \( d_{\min} \)? Ans: since \( \min \{ \| \tilde{c}_k \| \} = 2 \), \( d_{\min} = 2 \).
   - Q3: How many errors is this code guaranteed to correct/detect?
     - Ans: \( t_{\text{correct}} \leq d_{\min} - 1 = 1 \) \( t_{\text{detect}} \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor = 0 \).
3. For any linear code with Hamming distance at least $2t+1$,

$$2^{n-k} \geq 1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}$$

**Proof:** HD at least $2t+1$ (i.e. $d_{\text{min}} = 2t+1$)  
$\Rightarrow$ code can correct $t$ or fewer errors!

Number of situations of no error $= 1$

- 1-error $= \binom{n}{1}$
- 2-errors $= \binom{n}{2}$
- \vdots \hspace{1cm} \vdots \hspace{1cm} \vdots 
- $t$-errors $= \binom{n}{t}$

So that the total number of these situations is

$$1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}$$

- This code can distinguish all these situations.

On the other hand,

There are $2^{n-k}$ possible distinct parity-check combinations;

i.e. this code can distinguish at most $2^{n-k}$ situations,

So that

$$2^{n-k} \geq 1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}$$

as stated above.
Hamming Codes

Consider the \((7, 4)\) HC discussed in class, with the 3r parity equations:

\[
\begin{align*}
P_1 &= d_1 + d_2 + d_4 \\
P_2 &= d_1 + d_3 + d_4 \\
P_3 &= d_2 + d_3 + d_4
\end{align*}
\]

\(eqn \text{ set } 1\)

where all additions are in \(GF(2)\).

It can be seen that \(d_{\min} = 3\), as the parity calculations contribute at least a weight-2 word. Venn diagrams can be sketched to show which data bits are protected by which parity bit, as below.

The syndrome equations (showing error positions) are then given by \((eqn \text{ set } 2)\):

\[
\begin{align*}
E_1 &= d_1 + d_2 + d_4 + P_1 \\
E_2 &= d_1 + d_3 + d_4 + P_2 \\
E_3 &= d_2 + d_3 + d_4 + P_3
\end{align*}
\]

\(eqn \text{ set } 2\)

2) If \(E_3E_2E_1 = 101\), only \(d_2\) in error!
3) If \(E_3E_2E_1 = 001\), only \(P_1\) in error!

A syndrome table facilitates these diagnoses.