Revisit the matrix identity in the context of codes:
\[
\langle (I_k | A) \rangle = \langle (A^T | I_{nk}) \rangle^T
\]
Regarding the upper matrix as a systematic encoder for an \((n,k)\) code with the lower as the check matrix, the identity tells us that the check values are positioned opposite the weight-one columns (of \(H\)).

**Example:** \(H = \left( \begin{smallmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{smallmatrix} \right)\) has rank 3.

\(C = \langle H^\perp \rangle^\perp\) is a \((7,4,3)\) code.

\(a_1, a_2, a_3, a_4 \mapsto (a_1 + a_2 + a_4, a_1 + a_3 + a_4, a_1, a_2 + a_3 + a_4, a_2, a_3, a_4)\)

\((1001) \mapsto (0011001)\).

If \(\overline{e} = (0001001)\), then \(H^T \overline{e} = (011)\), which is taken as binary index of error with this formula for encoding:

\(H^T e = \begin{cases} \text{binary index of error, if wt}(e) = 1, \\ \text{if } e \notin \langle H^\perp \rangle^\perp \end{cases}\)

2. Variation on Hamming Code (See "shortened codes")

\(H = \left( \begin{smallmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{smallmatrix} \right)\) has rank 4 = \(4^\perp\) \((10,6,3)\) code.

\(a_1, a_2, a_3, a_4, a_5, a_6 \mapsto (a_1, a_2, a_3, a_4, a_5, a_6)\) data bits.

As before, \(H^T e = \begin{cases} (0000), \text{ if } e \notin \langle H^\perp \rangle^\perp, \\ \text{binary index of error, if } \text{wt}(e) = 1 \end{cases}\)

This \((10,6,3)\) code is not a perfect code.

If \(H^T e\) does not match any column of \(H^T\), then \(\text{wt}(e) > 1\).

**Example:** \(H^T e = (1101)\) must involve two or more errors:

\[
\begin{align*}
(00000011001) & \quad \text{Equally-likely weight-two errors with } H^T e = (1101) \\
(0001000010) & \\
(0000100100) &
\end{align*}
\]

Often, one declares "Decoding Failure" in such situations.
3. \(n-k\) "small" → build a syndrome table, size \(\frac{n-k}{n-k}\) (make decoding almost as simple as Hamming codes)

Suppose we have a check matrix \(H\) for \((n, k, d)\) linear code. Set \(\left\lfloor \frac{n-k}{2} \right\rfloor \leq t\) (correction capacity)

**Principle:** If \(t \geq \left\lfloor \frac{n-k}{2} \right\rfloor\), then \(H\overline{u}^T \neq H\overline{u}^T\)

This means we can construct a syndrome table by pairing each \(\overline{u}\) of weight \(\leq t\) with its syndrome \(H\overline{u}^T\). In effect, this assumes an all-0 codeword. [Reprove Hamming bound]

If \(d\) is unknown, build the table using one weight "class" at a time. When a "collision" occurs, the correction capacity is exceeded.

4. Technical property of syndrome table (systematic codes)

Suppose \(G = [I_K | A]\) associated with \((n, k, d)\) code

\(H = [-A^T | I_{n-k}]\)

Set \(t = \left\lfloor \frac{n-k}{2} \right\rfloor\), and assume \(\overline{e} = \overline{c} + \overline{e}\), with \(wt(\overline{e}) \leq t\)

Then data values in \(\overline{c}\) are correct if \(wt(H\overline{e}^T) \leq t\)

**Sketch of Proof:** Observe for \(\overline{e} = (e_0,\ldots, 0, e_{k+1}, e_{k+2},\ldots, e_n)\)

we have \(H\overline{e}^T = (e_{k+1}, e_{k+2},\ldots, e_n)\)

(Binary) # syndromes of weight \(\leq t\): \(\sum_{i=0}^{t} \binom{n-k}{i}\)

# \(\overline{e} = (e_0,\ldots, 0, e_{k+1},\ldots, e_n)\) of weight \(\leq t\): \(\sum_{i=0}^{t} \binom{n-k}{i}\)

The claim follows.
Start with an \((n, k, d)\) code \(C\) that allows simple decoding. A 2-D Product Code creates an \((n^2, k^2, d^2)\) which can be decoded by iterated application of the \((n, k, d)\) decoder. Expand a \(k \times k\) array of data values by encoding "across" and "down" to create a \(n \times n\) array (codeword)

\[
\begin{array}{cccc}
& & & \\
& & & \\
& & & \\
& & & \\
\end{array}
\]

transmit one row at a time, and store in a \(n \times n\) buffer.

For cases of practical interest, \(C\) is not perfect

**ONE**
- Decade columns, skipping "Decoding Failures"

**ITERATION**
- Decade rows, skipping "Decoding Failures"

The number of iterations required is determined empirically.

Often the minimum distance \(d^2\) significantly underestimates number of correctable errors.