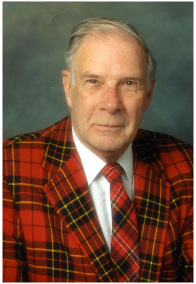
Error-correcting codes

- Originally inspired by errors in reading programs on punched cards
- Now used in WiFi, cell phones, communication with satellites and spacecraft, digital television, RAM, disk drives, flash memory, CDs, and DVDs

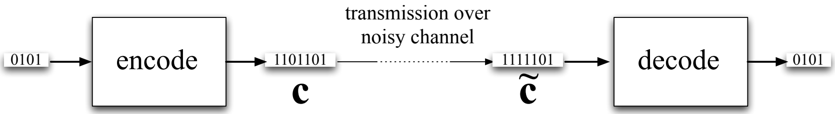


Richard Hamming

Hamming code is a *linear binary block code*:

- *linear* because it is based on linear algebra,
- *binary* because the input and output are assumed to be in binary, and
- *block* because the code involves a fixed-length sequence of bits.

Error-correcting codes: Block codes



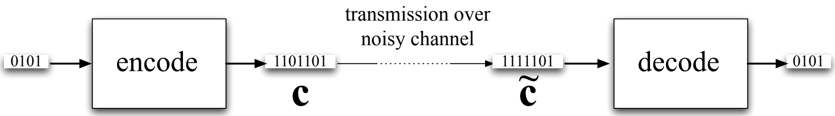
To protect an 4-bit block:

- Sender *encodes* 4-bit block as a 7-bit block $\mathbf{c}$
- Sender transmits $\mathbf{c}$
- $\mathbf{c}$ passes through noisy channel—errors might be introduced.
- Receiver receives 7-bit block $\tilde{\mathbf{c}}$
- Receiver tries to figure out original 4-bit block

The 7-bit encodings are called *codewords*.

$\mathcal{C}$ = set of permitted codewords

Error-correcting codes: Linear binary block codes



Hamming's first code is a *linear* code:

- Represent 4-bit and 7-bit blocks as 4-vectors and 7-vectors over $GF(2)$.
- 7-bit block received is $\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{e}$
- $\mathbf{e}$ has 1's in positions where noisy channel flipped a bit
  ($\mathbf{e}$ is the *error vector*)
- *Key idea:* set $\mathcal{C}$ of codewords is the null space of a matrix H.

This makes Receiver's job easier:

- Receiver has $\tilde{\mathbf{c}}$, needs to figure out $\mathbf{e}$.
- Receiver multiplies $\tilde{\mathbf{c}}$ by H.

$$H * \tilde{\mathbf{c}} = H * (\mathbf{c} + \mathbf{e}) = H * \mathbf{c} + H * \mathbf{e} = \mathbf{0} + H * \mathbf{e} = H * \mathbf{e}$$

- Receiver must calculate $\mathbf{e}$ from the value of $H * \mathbf{e}$. *How?*

Hamming Code

In the Hamming code, the codewords are 7-vectors, and

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

Notice anything special about the columns and their order?

- Suppose that the noisy channel introduces at most one bit error.
- Then $\mathbf{e}$ has only one 1.
- Can you determine the position of the bit error from the matrix-vector product $H * \mathbf{e}$?

Example: Suppose $\mathbf{e}$ has a 1 in its third position, $\mathbf{e} = [0, 0, 1, 0, 0, 0, 0]$.

Then $H * \mathbf{e}$ is the third column of H, which is $[0, 1, 1]$.

As long as $\mathbf{e}$ has at most one bit error, the position of the bit can be determined from $H * \mathbf{e}$. This shows that the Hamming code allows the recipient to correct one-bit errors.

Hamming code

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

Quiz: Show that the Hamming code does not allow the recipient to correct two-bit errors: give two different error vectors, $\mathbf{e}_1$ and $\mathbf{e}_2$, each with at most two 1's, such that $H * \mathbf{e}_1 = H * \mathbf{e}_2$.

Hamming code

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

Quiz: Show that the Hamming code does not allow the recipient to correct two-bit errors: give two different error vectors, $\mathbf{e}_1$ and $\mathbf{e}_2$, each with at most two 1's, such that $H * \mathbf{e}_1 = H * \mathbf{e}_2$.
Answer: There are many acceptable answers. For example, $\mathbf{e}_1 = [1, 1, 0, 0, 0, 0, 0]$ and $\mathbf{e}_2 = [0, 0, 1, 0, 0, 0, 0]$ or $\mathbf{e}_1 = [0, 0, 1, 0, 0, 1, 0]$ and $\mathbf{e}_2 = [0, 1, 0, 0, 0, 0, 1]$.