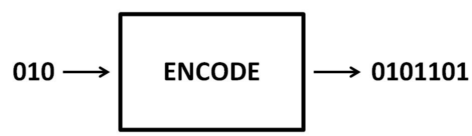
EECS 16A Designing Information Devices and Systems I
Spring 2015 **Discussion 4W**

Hamming Code

Hamming code is the first type of *error-correcting codes*, and is still used in transmission (e.g. WiFi) and storage (e.g. flash memory). It is a *linear binary block code* that relies on linear algebra to correct errors in blocks of binary data.

Encoding

A block of bits is represented by a vector over binary field. With the Hamming code, we want to transmit a block of length 3. The first (error-less) step is to encode the bits we want to transmit to a binary block of length 7 (another vector over binary field). The resulting 7 bit block is called a *codeword*.



Noisy Channel

This codeword is then passed through a noisy channel that can introduce errors. Let $\mathbf{c}$ and $\tilde{\mathbf{c}}$ denote the codeword that goes into the noisy channel and the binary block that comes out, respectively.

Error Check

The error check step is essentially a matrix multiplication. A matrix H is used to transform the binary block, $\tilde{\mathbf{c}}$, that comes out of the noisy channel. For Hamming code,

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

The final result is $H\tilde{\mathbf{c}}$. The key to this coding algorithm is that **all codewords c are of a form such that $\mathbf{Hc} = \mathbf{0}$**. Therefore, if $H\tilde{\mathbf{c}} \neq 0$, then we know that some error was introduced via the noisy channel.

1. Subspace of Codewords

Given the property that $H\mathbf{c} = 0$, prove that the set of all such codewords (or vectors) $\mathbf{c}$ forms a subspace.

2. Detecting Error Location

Say we find that $H\tilde{\mathbf{c}} \neq 0$. We know that there is some error, but we don't know where (which bit) this error is. We can model the equation $H\tilde{\mathbf{c}}$ as $H(\mathbf{c}+\mathbf{e})$, where $\mathbf{c}$ is the correct codeword and $\mathbf{e}$ is the error that was introduced.

(a) In case of an error, $H\tilde{\mathbf{c}} = H\mathbf{e}$. Why?

(b) Suppose $\mathbf{e} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$. What would the result of $H\tilde{\mathbf{c}}$ be?

(c) Say $H\tilde{\mathbf{c}} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$. Can you determine which bit the error was in (what is $\mathbf{e}$)?

(d) What do you notice about the structure (and the individual columns) of H? How is this useful for error detection?

(e) Is it possible to detect the position of two errors? How about three? Why or why not?
Hint: Create error vectors $\mathbf{e}$ *that have* $1s$ *in more than one position. Multiply by the* H *matrix and try to determine the location of error.*

3. 2×2 Gaussian Elimination

Invert the following matrix using Gaussian elimination: $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$