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.

\[
\begin{array}{c}
010 \\
\end{array} \rightarrow \text{ENCODE} \rightarrow \begin{array}{c}
0101101 \\
\end{array}
\]

Noisy Channel

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

\[
\begin{array}{c}
0101101 \\
\end{array} \rightarrow \text{NOISY CHANNEL} \rightarrow \begin{array}{c}
0101001 \\
\end{array}
\]

Error Check

The error check step is essentially a matrix multiplication. A matrix \( H \) is used to transform the binary block, \( \tilde{c} \), 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 \( H\tilde{c} \). The key to this coding algorithm is that all codewords \( c \) are of a form such that \( Hc = 0 \). Therefore, if \( H\tilde{c} \neq 0 \), then we know that some error was introduced via the noisy channel.
1. Subspace of Codewords

Given the property that $Hc = 0$, prove that the set of all such codewords (or vectors) $c$ forms a subspace.

2. Detecting Error Location

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

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

(b) Suppose $e = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T$. What would the result of $H\tilde{c}$ be?

(c) Say $H\tilde{c} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$. Can you determine which bit the error was in (what is $e$)?
(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 $e$ 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: $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$