Binary Hamming Codes

Hamming codes were discovered by R.W. Hamming and M. J. E. Golay. They form an important class of codes – they have interesting properties and are easy to encode and decode.

While Hamming codes are defined over all finite fields, here, we only discuss the binary Hamming codes.

Binary Hamming codes are a family of binary linear error-correcting codes that can detect up to two-bit errors or correct one-bit errors. For each integer $m > 2$, there is a

$$[\,2^m - 1,\ 2^m - m - 1,\ 3\,] \quad \text{Hamming code.}$$

This implies that all Hamming codes have a minimum distance of 3, which means that the code can detect and correct a single error and detect double-bit errors. By including an extra parity bit, it is possible to increase the minimum distance of the Hamming code to 4. This gives the code the ability to detect up to 3 errors but not correct any. Because of the simplicity of Hamming codes, they are popular in computer memory systems (RAM, Random Access Memory), where they are known as *SECDED* ("Single Error Correction, Double Error Detection"). Particularly popular code is the $[\,72, 64\,]$ code, a truncated $[\,127, 120\,]$ Hamming code plus an additional parity bit.

The parity-check matrix of a Hamming code is constructed by listing all rows of length m, where each row is a binary representation of a number from 1 to $2^m - 1$ (not particularly in ascending or descending order). The parity-check matrix has the property that any two columns are pairwise linearly independent.

- The dual code of the Hamming code is the punctured Hadamard code.

Although any number of algorithms can be created, the following general algorithm positions the parity bits at powers of two to ease calculation of which bit was flipped upon detection of incorrect parity.
I. All bit positions that are powers of two are used as parity bits.

$$\text{positions:}\ \ 1,\ 2,\ 4,\ 8,\ 16,\ 32,\ 64,\ \ \text{etc.}$$

II. All other bit positions are for the message to be encoded.

$$\text{positions:}\ \ 3,\ 5,\ 6,\ 7,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14,\ 15,\ 17,\ \ \text{etc.}$$

III. Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.

♣ **[7,4,3] Hamming Codes**. In a 7-bit message, there are seven possible single bit errors, so three error control bits could potentially specify not only that an error occurred but also which bit caused the error. In Shannon's paper, the following algorithm (due to Richard Hamming) describes a $[7,\ 4,\ 3]$ binary Hamming code together with its decoding scheme: Let $v = (v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6,\ v_7) \in \mathbb{K}^7$, where $v_3,\ v_5,\ v_6,\ v_7$ are message bits and $v_1,\ v_2,\ v_4$ are parity bits calculated as follows:

1. v_1 is chosen to make $\alpha = v_1 + v_3 + v_5 + v_7 \equiv 0 \quad mod\ 2$.
2. v_2 is chosen to make $\beta = v_2 + v_3 + v_6 + v_7 \equiv 0 \quad mod\ 2$.
3. v_4 is chosen to make $\gamma = v_4 + v_5 + v_6 + v_7 \equiv 0 \quad mod\ 2$.

Notice that α and β share v_3 and v_7, α and γ share v_5 and v_7, and β and γ share v_6 and v_7. When a word is received, then $\alpha,\ \beta$, and γ are calculated; zero indicates no error occurred and one gives the v_i that is incorrect.

Here is a systematic generating matrix of the $[7,\ 4,\ 3]$ Hamming code using the above algorithm:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

It generates the following Hamming code:

$$C = \left\{ \begin{array}{cccccccc} 0000000 & 1110000 & 1001100 & 0101010 & 1101001 & 0111100 & 1100110 & 1000011 \\ 0011001 & 1011010 & 0100101 & 0010110 & 1010101 & 0110011 & 0001111 & 1111111 \end{array} \right\}$$

Here is how to correct errors for word received by any systematic Hamming code using the above algorithm, assuming that only one error bit occurred:

$(\alpha,\ \beta,\ \gamma)$	*error* $\longleftrightarrow$ *bit*
$1 \longleftrightarrow (1,\ 0,\ 0)$	$1000000 \longleftrightarrow\ 1$
$2 \longleftrightarrow (0,\ 1,\ 0)$	$0100000 \longleftrightarrow\ 2$
$4 \longleftrightarrow (0,\ 0,\ 1)$	$0001000 \longleftrightarrow\ 4$
$3 \longleftrightarrow (1,\ 1,\ 0)$	$0010000 \longleftrightarrow\ 3$
$5 \longleftrightarrow (1,\ 0,\ 1)$	$0000100 \longleftrightarrow\ 5$
$6 \longleftrightarrow (0,\ 1,\ 1)$	$0000010 \longleftrightarrow\ 6$
$7 \longleftrightarrow (1,\ 1,\ 1)$	$0000001 \longleftrightarrow\ 7$

Example. In all that follows, we can see how the above decoding scheme detects, corrects, and decodes words:

Error :	*Zero Error*	*One Error*	*Two Errors*	*Three Errors*	*Four Errors*
Sent :	$v_0 = 0001111$	$v_1 = 0011001$	$v_2 = 1010101$	$v_3 = 0111100$	$v_4 = 1001100$
Received :	$w_0 = 0001111$	$w_1 = 0011011$	$w_2 = 0011101$	$w_3 = 1001100$	$w_4 = 1000011$
$(\alpha\ \beta\ \gamma)$:	$(0\ 0\ 0) \longleftrightarrow 0$	$(0\ 1\ 1) \longleftrightarrow 6$	$(1\ 0\ 1) \longleftrightarrow 5$	$(0\ 0\ 0) \longleftrightarrow 0$	$(0\ 0\ 0) \longleftrightarrow 0$
Decoded :	$v_0' = 0001111$	$v_1' = 0011001$	$v_2' = 0011001$	$v_3' = 1001100$	$v_4' = 1000011$
Detected :	*Yes*	*Yes*	*Yes*	*No*	*No*
Corrected :	*Yes*	*Yes*	*No*	*No*	*No*

Notice that the received word w_2 with two error-bits was decoded as v_2', but $v_2' \notin C$, so we know that v_2' is not the right word and it is impossible for us to guess the right codeword. An interesting fact about this algorithm is that a received word can be decoded without the use of a parity check matrix or coset leaders.

Exercise. Use the above decoding scheme to detect and/or correct the following received words:

$$\textit{Received}: \quad w_0 = 0111100 \quad w_1 = 1001010 \quad w_2 = 0100111 \quad w_3 = 0010111 \quad w_4 = 0001110.$$

One possibility for a parity-check matrix H_s and a generator matrix G_s for a [7,4,3] Hamming code is:

$$H_s = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Notice that G_s is in standard form. By permuting columns of G_s or rows of H_s, we obtain systematic Hamming code. We shall see cyclic Hamming codes with non-systematic generator.

♣ **Binary Linear Cyclic Hamming Codes**. The $[7,\ 4,\ 3]$ binary cyclic code generated by the polynomial $g(x) = 1 + x + x^3$ is a Hamming code:

$$H_c = \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \\ 1\ 1\ 0 \\ 0\ 1\ 1 \\ 1\ 1\ 1 \\ 1\ 0\ 1 \end{bmatrix} \quad \text{with} \quad G_c = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

This generator matrix is not in a systematic form and does not use the previous algorithm. The encoding and decoding algorithms for Hamming cyclic codes are based on multiplication and division of polynomials which is discussed in the binary cyclic code section.

Notice that

$$[2^4 - 1,\ 2^4 - 4 - 1,\ 3] = [15,\ 11,\ 3]\,;$$

therefore the binary linear cyclic code in $\mathbb{K}^{15}$, generated by the polynomial $g(x) = 1+x+x^4$ must be a Hamming code.

♣ **Using Dual Code to Obtain the Parity-check Matrix of a Hamming code.** The fact that a Hamming code C is also a cyclic code, we may use the transpose of the generator matrix of $C^{\perp}$ as the parity-check matrix of the code C.

Consider the binary $[7,4,3]$- Hamming code with the generator polynomial $g(x)=1+x+x^3$ and generator matrix

$$G=\begin{bmatrix} g(x)\\ x\,g(x)\\ x^2\,g(x)\\ x^3\,g(x)\end{bmatrix}=\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{bmatrix}.$$

We have

$$C=\begin{Bmatrix}0000000&1101000&0110100&0011010&0001101&1011100&1110010&1100101\\0101110&0111001&0010111&1000110&0100011&1001011&1010001&1101000\end{Bmatrix}$$

The generator polynomial of $C^{\perp}$ is $h(x)=1+x^2+x^3+x^4$ and its generator matrix is

$$G^{\perp}=\begin{bmatrix} h(x)\\ x\,h(x)\\ x^2\,h(x)\end{bmatrix}=\begin{bmatrix}1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}.$$

By transposing the generator matrix of $C^{\perp}$, we may obtain a parity-check matrix for the code C. Thus

$$H=\left(G^{\perp}\right)^t=\begin{bmatrix}1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}^t=\begin{bmatrix}1&0&0\\0&1&0\\1&0&1\\1&1&0\\1&1&1\\0&1&1\\0&0&1\end{bmatrix}.$$

If $w=0\,0\,1\,1\,0\,0\,0\leftrightarrow w(x)=x^2+x^3$ is received, then

$$s_0(x)=s(x)\equiv w(x)\ mod\ g(x)\equiv x^2+x^3\ mod\ (1+x+x^3)\equiv 1+x+x^2$$

is the syndrome polynomial. We next compute

$$s_1(x)\equiv xs(x)\ mod\ g(x)\equiv x(1+x+x^2)\ mod\ g(x)\equiv 1+x^2$$
$$s_2(x)\equiv x^2s(x)\ mod\ g(x)\equiv x(1+x^2)\ mod\ g(x)\equiv 1$$

which has weight $t=1$. So $j=2$ and therefore

$$u(x)=x^{7-2}s_2(x)\ mod\ (1+x^7)\equiv x^5.$$

Thus

$$v(x)=w(x)+u(x)=(x^2+x^3)+x^5\implies 0\,0\,1\,1\,0\,1\,0$$

is the most likely codeword.

The product of w by H produces the syndrome $0\,1\,1$, which is the sixth row of $H\leftrightarrow\ x^5$. Hence

$$v(x)=w(x)+x^5\implies v=0\,0\,1\,1\,0\,0\,0+0\,0\,0\,1\,0\,0\,0=0\,0\,1\,1\,0\,1\,0$$

is the most likely codeword.

Exercise. Consider the binary $[7,4,3]$- Hamming code with the generator polynomial $g(x) = 1 + x^2 + x^3$.

Suppose the following words are received:

$$\textit{Received}: \quad w_0 = 0111100 \quad w_1 = 1001010 \quad w_2 = 0100111 \quad w_3 = 0010111 \quad w_4 = 0001110\ .$$

(a) Use the polynomial syndrome to decode the above received words:

(b) Use the generator matrix of $C^{\perp}$ to decode the above received words:

♣ **Extended Hamming Codes**. An extension of a binary Hamming code results from adding at the beginning or at the end of each codeword a new digit that checks the parity of the codeword. Therefore, every word in an Extended Hamming code has an even number of ones. This way, the minimum distance of the Hamming code is increased from 3 to 4. This gives the code the ability to detect and correct a single error and it could also be used to detect up to 3 errors but not correct any.

It is important to note that by extending a code, we increase the length of the codewords but we keep the dimension of the code.

By adding a parity bit at the end of our first [7,4,3] Hamming code, we obtain an [8,4,4] Extended code which changes the generator matrix G into $\widehat{G}$:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \widehat{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

[7,4,3] Hamming Code [8,4,4] Extended Hamming Code

• An [8,4,4] Extended Hamming code is a self-dual $\mathcal{RM}\,(1,3)$ Reed-Muller code.

The following Extended Hamming code generated by $\widehat{G}$ still has sixteen codewords:

$$\widehat{C} = \left\{ \begin{array}{cccccccc} 00000000 & 11100001 & 10011001 & 01010101 & 11010010 & 01111000 & 11001100 & 10000111 \\ 00110011 & 10110100 & 01001011 & 00101101 & 10101010 & 01100110 & 00011110 & 11111111 \end{array} \right\}$$

• Due to the limited redundancy that Hamming codes add to the data, they can only detect and correct errors when the error rate is low. This is the case in computer memory (ECC memory), where bit errors are extremely rare and Hamming codes are widely used. In this context, an extended Hamming code having one extra parity bit is often used. Extended Hamming codes achieve a Hamming distance of 4, which allows the decoder to distinguish between when at most one 1-bit error occurs and when any 2-bit errors occur. In this sense, extended Hamming codes are single-error correcting and double-error detecting, abbreviated as SECDED.