A Note on Hamming Code

The Hamming code is a powerful error correcting code. It enables us to detect errors and to recover the original binary word if one digit goes wrong.

Let $s = (s_1, s_2, s_3, s_4)$ be a 4-long binary word (i.e., every s_j is either 0 or 1). The *Hamming code*, $H(s)$, of s is a 7-long word defined as follows:

$$\begin{aligned}
H(s)_1 &= H(s)_3 + H(s)_5 + H(s)_7 \pmod 2 = s_1 + s_2 + s_4 \pmod 2 \\
H(s)_2 &= H(s)_3 + H(s)_6 + H(s)_7 \pmod 2 = s_1 + s_3 + s_4 \pmod 2 \\
H(s)_3 &= s_1 \\
H(s)_4 &= H(s)_5 + H(s)_6 + H(s)_7 \pmod 2 = s_2 + s_3 + s_4 \pmod 2 \\
H(s)_5 &= s_2 \\
H(s)_6 &= s_3 \\
H(s)_7 &= s_4
\end{aligned}$$

For example, if $s = (1, 0, 0, 0)$, then $H(s) = (1, 1, 1, 0, 0, 0, 0)$. Indeed,

$$\begin{aligned}
H(s)_1 &= H(s)_3 + H(s)_5 + H(s)_7 \pmod 2 = s_1 + s_2 + s_4 \pmod 2 = 1 \\
H(s)_2 &= H(s)_3 + H(s)_6 + H(s)_7 \pmod 2 = s_1 + s_3 + s_4 \pmod 2 = 1 \\
H(s)_4 &= H(s)_5 + H(s)_6 + H(s)_7 \pmod 2 = s_2 + s_3 + s_4 \pmod 2 = 0.
\end{aligned}$$

In general, we will refer to the bits coming from the original binary word s as data bits, and the rest as parity bits. In the above example, the parity bits are the first two occurrences of 1 and the first occurrence of 0.

Let t be a 7-long binary word such that

- there is no 4-long binary word s for which $t = H(s)$,
- if we change a certain bit in t, then it becomes the Hamming code for some 4-long binary word u.

We claim that u can be recovered from t.

By assumption there is precisely one incorrect bit in t, but we do not know which one. Consider the following algorithm. Let the sequence v consist of the data bits of t and its Hamming code be $H(v)$. There are two cases.

Case 1: one data bit t_i is incorrect. Then some of the parity bits will be different in t and in $H(v)$. Looking at the definition of the parity bits we can figure out which data bit t_i is incorrect. Note also that there are more than one parity bits in t and $H(v)$ which disagree.

Case 2: one parity bit t_i is incorrect. Then all the other parity bits are correct. Thus changing t_i in t yields a binary word such that it is the Hamming code of v.

Finally note that cases 1 and 2 can be distinguished by the number of parity bits that differ in t and in $H(v)$. Thus we know which one of the cases apply.

As an example let us look at the binary word $t = (1, 1, 1, 0, 0, 1, 0)$. Then $v = (1, 0, 1, 0)$ and $H(v) = (1, 0, 1, 1, 0, 1, 0)$. Hence the disagreeing parity bits are $t_2 = 1 \neq 0 = H(v)_2$ and $t_4 = 0 \neq 1 = H(v)_4$. Thus case 1 above applies and we conclude that we should change $t_{2+4} = t_6$.

Indeed, if you take the sequence $t = (1, 1, 1, 0, 0, 0, 0)$, it turns out to be the Hamming code of $(1, 0, 0, 0)$.

Now consider the binary word $t = (1, 0, 1, 0, 0, 0, 0)$. Then $v = (1, 0, 0, 0)$ and $H(v) = (1, 1, 1, 0, 0, 0, 0)$. Thus case 2 applies, and we get the correct Hamming code by changing the second bit in t.