Linear Codes

Example: Hamming $[7,4]$ code ($n=7$, $k=4$)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \in \{0,1\}^{k \times n}$$ encoding matrix

Encoding

Suppose Alice wants to send Bob a 4-bit original message $v \in \{0,1\}^4$

Alice sends $((v \cdot G) \bmod 2) \in \{0,1\}^7$ as the encoding of v.

Say $v = (1100)$

$v \cdot G$ is the sum of first and second row of G.

$$v \cdot G \equiv (1000110) + (0100101) \pmod 2$$
$$\equiv (\underline{1100}011)$$

First 4 bits are always the original message.

How about decoding?

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \in \{0,1\}^{(n-k) \times n}$$ decoding matrix.

Suppose we received an (possibly corrupted) encoding

$c \in \{0,1\}^n$

We compute $c \cdot H^T$, if it is equal to the k-th column of H, we conclude the k-th bit is flipped.

Example 1: $c = (1100001)$

$c \cdot H^T = (110) + (101) + (001)$
$= (010)$ which is the 6-th column of H $\Rightarrow$ The decoding algorithm concludes that the 6 th bit is flipped. $11000\underline{1}1 \Rightarrow v = 1100$.

Example 2: $c = (1\underline{0}00011) \Rightarrow c \cdot H^T = (101) =$ 2nd column of H

correct $c = 1100011 \Rightarrow v = 1100$

Example 3: $c = (1100011) \Rightarrow c \cdot H^T = (000) =$ no error.

Remark: The Hamming $[7,4]$ code can correct up to 1 bit of error.

More efficient than 3-repetition code $(7 < 12)$

Linear Codes

A subspace of $\{0,1\}^n$ is a set S of vectors such that $\forall s_1, s_2 \in S$, $(s_1 + s_2) \bmod 2$ is also in S.

(The codewords of) An $[n,k]$-linear code is a k-dimensional subspace of $\{0,1\}^n$.

- original message has length k
- encoding has length n.
- if c_1 and c_2 are both codewords then so is $(c_1 + c_2) \bmod 2$.

Examples.
- 3-repetition
- parity check
- 2D parity check
- [7,4] Hamming

These are all linear codes.

(We can write down the set of codewords and verify they form a subspace.)

We can quickly compute the minimum distance of a linear code.

Lemma: The minimum distance of a linear code C is equal to the minimum Hamming weight of all non-zero codewords.
(i.e., # of 1's)

($c_1 + c_1 = 0$, so for linear codes, $(00\ldots0)$ is always a codeword).

Proof: Let u be a codeword with minimum Hamming weight. $wt(u)$

$d(C) \leq wt(u)$: because $wt(u) = d(u, 0)$ is the distance between two codewords.

$d(C) \geq wt(u)$: for any two codewords v and w. $(v-w)$ is a codeword.

$$d(v, w) = wt(v-w) \geq wt(u)$$

Therefore, $d(C) = wt(u)$

Construction.

To construct an $[n, k]$ linear code, we need to construct a k-dimensional subspace of $\{0,1\}^n$.

The easiest way is to pick k independent vectors and take their span.

The encoding / generating matrix G is a $k \times n$ matrix formed by these vectors (as rows).

Linear independent

A set S of vectors is linear independent iff one cannot express any vector in S as a linear combination of other vectors in S.

Example 1: a $[6,2]$ linear code with

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

$k=2$, $n=6$

The set of codewords $= \{ \underset{\updownarrow}{\overset{00}{}} 000000,\ \overset{10}{101010},\ \overset{01}{010101},\ \overset{11}{111111} \}$.

$$\begin{array}{cccc} 00 & 10 & 01 & 11 \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 000000 & 101010 & 010101 & 111111 \end{array}$$

this must be a codeword by the definition of linear codes

Example 2: a $[5,4]$ linear code with

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

This is in fact a parity check code

Decoding: If the encoding matrix $G = [\, I_k \quad P \,]$ then the decoding matrix $H = [-P^T \quad I_{n-k}]$ $= [\, P^T \quad I_{n-k} \,]$ for binary code.

(I_d = the $d \times d$ identity matrix)

Example 1: $H = \begin{pmatrix} 1 & 0 & 1 & & & \\ 0 & 1 & & 1 & & \\ 1 & 0 & & & 1 & \\ 0 & 1 & & & & 1 \end{pmatrix}$

Example 2: $H = (1\ 1\ 1\ 1\ 1)$.

Error detection

Theorem. $vH^T = 0$ iff $v \in C$.

(proved on page 411 of the textbook)

Error correction.

Let v be the received encoding (possibly corrupted).

Algorithm 1: find the codeword $c \in C$ that is closest to v.

Simple. but slow!

Algorithm 2:

First list all codewords on the first row.

Then among the remaining vectors, choose one with smallest Hamming weight, add this vector to each vector in the first row to obtain the second row.

Finally we decode by changing the received vector to the one at the top of its column.

Example: $G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$

$v = 1001.$

error detection: $H = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$

$vH^T = (10) \neq (00)$, so $v \notin C$.

error correction.

1011 + 0110.

0000	1011	0110	1101	← all codewords
1000	0011	1110	0101	← first row + 1000
	$\overset{1011 + 1000}{\|\|}$			
0100	1111	0010	1001	
0001	1010	0111	1100	

So $1001 \to 1101 \to 11$. = the original message.