

## ERROR-CORRECTING CODES - WEEK 2

### Hamming Codes and Perfect Codes

The standard array of an  $(n,k)$  binary code has its first row formed by  $2^k$  codewords. The next  $nC1 (=n)$  rows are cosets with coset leaders the  $n$  single-bit errors. Cosets are formed by choosing as coset leader an  $n$ -vector not already written down, so we may continue adding  $nC2$  rows with all 2-bit errors as coset leaders. There will be no clash if the distance of the code is 5 or more, because no  $n$ -vector in the array could simultaneously be at distance 2 from two distinct codewords. We may continue thus with 3-bit errors as coset leaders ( $nC3$  of them) provided we do not exceed the maximum total of  $2^{(n-k)}$  rows. Thus a  $(15,7)$  code with  $2^7$  codewords can have at most  $2^8 = 256$  rows. Now  $1 + nC1 + nC2 = 121$  with  $n=15$  so a distance-2  $(15,7)$  code appears possible, but a distance-3  $(15,7)$  code is not; because  $1 + nC1 + nC2 + nC3 = 455 > 256$ .

*Question: is a distance-3  $(15,5)$  code possible?*

When  $\sum([nCj] \text{ for } j=1,t) = 2^{(n-k)}$  then we could have a **perfect** distance- $t$  code. The standard array based on coset leaders of up to  $t$  bit-errors fits exactly into the space available. **Hamming Codes** are perfect codes. They have  $t=1$  and  $2^{(n-k)} = 1+n$ . Possibilities are:

$(n-k) = 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad \dots$   
 $n = 3 \quad 7 \quad 15 \quad 31 \quad 63 \quad \dots$  giving  $(3,1), (7,4), (15,11), (31,26), (63,57)$  Hamming Codes.

*Question: The  $(23,12)$  Golay Code is a Perfect Code. How many error bits should it be capable of correcting?*

### The Varsharmov-Gilbert Bound

The VG Bound is a lower bound on a graph relating code rate to distance. Below the VG bound, codes with paired (code rate – distance) values can always be found. This is in contrast to the sphere-packing bound which is an upper bound – no pairs exist above it. See Annexe 2.1. The VG bound is found by considering a given  $(n-k)$  (the number of rows in  $\mathbf{H}$ ) and adding linearly independent columns (whose number controls the distance), thereby increasing  $n$  and the code rate ( $k/n = 1 - (n-k)/n$ ) for some starting value of  $d$ , until it is no longer possible. Adding a new column to  $\mathbf{H}$  can never increase the distance –

*Question: Why?*

Suppose we already have  $(n-1)$  columns subject to the constraint that any  $j$  columns ( $j \leq 2t$ ) are linearly independent. Choose and add a new column that is not equal to any linear combination of  $(2t-1)$  existing columns. Now no  $2t$  columns are linearly dependent. The choice is always possible if all linear combinations of existing ones is such that:

$\sum [(n-1)Cj, j=1, (2t-1)] < 2^{(n-k)} - 1$ , the number of possible non-zero columns. For large  $n$  this gives:  
 $2^{((n-1)H(m))} < 2^{(n-k)}$  with  $m = (2t-1)/(n-1)$ .

So for large  $n$  we can create a code of distance  $(2t+1)$  provided  $n.H(2t/n) < (n-k)$  or  $k/n < 1 - H(2t/n)$ . **This is the VG Bound .**

### The Plotkin Bound

The Plotkin Bound is an upper bound for the existence of linear codes.

In any given bit position half the codewords have 0, half have 1. *Exercise: Students to show this by considering cosets. Students to show similar result for non-binary codes.*

- 1) Given a length  $n$  and distance  $d=2t+1$  consider that code which has maximum number of codewords  $M(n,t)$  with those parameters. The set of all codewords ending in 0 is a subspace holding half the codewords. Therefore  $M(n-1,t) \geq M(n,t)/2$ , or  $M(n,t) \leq 2.M(n-1,t)$
- 2) The total weight of the code is  $n.2^{k-1}$ . The distance  $d$ , the minimum weight, is less than the average weight of non-zero codewords, so  $d < \frac{n.2^{k-1}}{2^k - 1}$ . This simplifies (for the optimal code) to  $M(n,t) = 2^k \leq \frac{2d}{2d-n}$ , provided that  $2d-n$  is positive..
- 3) For a given  $d$  consider that  $n^* = 2d-1$ , so  $M(n^*,t) \leq 2d$ . For larger  $n$  we get  $2^k = M(n,t) \leq 2d.2^{n-n^*} = d.2^{n+2-2d}$ . So  $k \leq n+2-2d + \log d$ . Approximating then gives the **Plotkin Bound for code rate**  $k/n \leq 1 - 4t/n$ . See Annexe 2A.

### Modulation

Data are seldom transmitted as rectangular bits. The bits are encoded as continuous waveforms suitable to the medium (wire, fibre, radio etc.) usually employing a carrier signal. Even the old-fashioned copper telephone pair was used to send data encoded as frequencies – 980Hz representing 1 and 1180Hz representing 0. This is known as **FSK or Frequency Shift Keying**, and signalling rates up to 300 bps could be achieved. The modulator/demodulator is called a **modem**. The V26 modem encoded two bits, worked at 1200 symbols/per/second (baud) giving 2400 bits per second. **Differential Phase Shift Keying (D-4PSK)** was used, with an 1800Hz carrier and *changes* in data encoded by phase shifts: 00(+0degrees), 01(+90), 11(+180), 10(+270). (See Annexe 2B.) Note the use of Gray Coding in which the adjacent symbols on the phase diagram differ by only a single bit – *Why?*

*Exercise: Gray code the symbols for 8PSK.*

Much more sophisticated modems, encoding up to five bits per symbol, are now common. They employ both phase and amplitude modulation – and use TCM (Trellis Code Modulation).

The demodulator may misread a received symbol and so deliver the wrong bit-pattern. Misreading is typically due to noise on the channel. Noise is usually modelled as **AWGN – Additive White Gaussian Noise**. It affects the entire frequency range uniformly, has zero mean, a Normal distribution with a standard deviation expressing the strength of the noise, and is deemed to be added to the users' signal. One can calculate the resultant bit-error probability ( $p$ ) at the demodulator for each modulation technique as a function of the **Signal-Noise Ratio, SNR (S/N =A, say)**.

Typically  $p$  is proportional to  $\text{erfc}(rA^{1/2})$  or to  $e^{(-rA)}$ , for some constant  $r$ , where  $\text{erfc}(x)$  is the integral from  $x$  to infinity of  $e^{-t^2}.dt$  – that is, the tail ends of decaying exponential functions giving an ever

smaller  $p$  as SNR,  $A$ , increases. Note that  $p$  is the symbol error probability, but if Gray coding is used this may be taken to be the bit error probability. Annex 2C shows the dependence of  $p$  on SNR for some common modulation techniques.

[Capacity,  $C$ , previously defined in terms of discrete binary transmission of information can also be analysed in terms of continuous transmission using waveforms of appropriate frequencies. Shannon's Theorem shows that  $C = W \log_2 [ 1 + S/N ]$  bits per second where  $W$  is the bandwidth of the channel.]

### **Erasures**

Demodulation turns noisy waveforms into bits and in so doing interprets the received waveforms. Hard decision demodulation decides which waveforms were sent and outputs their digital values unqualified, and possibly erroneous. Instead of labelling each bit 1 or 0 a third possibility is to label it ?, signifying 'dubious' – an 'erasure'. Ordinary errors are unknown as to location and value; erasures' locations are known – for example scratches or dirt on a recording medium. If the distance of a binary code is  $2t+1$  and there are  $s$  errors and  $k$  erasures correct decoding is possible provided that  $2s + k \leq 2t$ . Suppose in reality that the erasures correspond to  $r$  1s and  $(k-r)$  0s. Set all the erasures to 0, giving  $s+r$  errors. Then set all the erasures to 1 giving  $s+k-r$  errors. Both of these expressions cannot simultaneously be greater than  $t$ , so one or the other will lead to successful decoding.

*Exercise: Explain why, and describe how used in practice.*