Error correcting codes

Michael Mo

26 January 2016

Uses of ECC

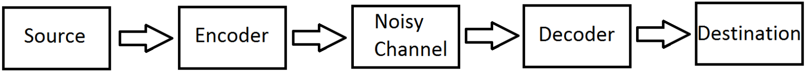
Where are error correcting codes needed?

- Computer memory
- CD's, DVD's
- Communication from earth to a satellite

Setting and model

Normal setting:

Setting with an encoding and decoding scheme:



Setting and model

Assumptions

- Binary bitstream
- Binary symmetric channel
- Noise only introduces bitflips

Block code

Encoder encodes fixed number of data bits every time and gives a fixed length binary string.

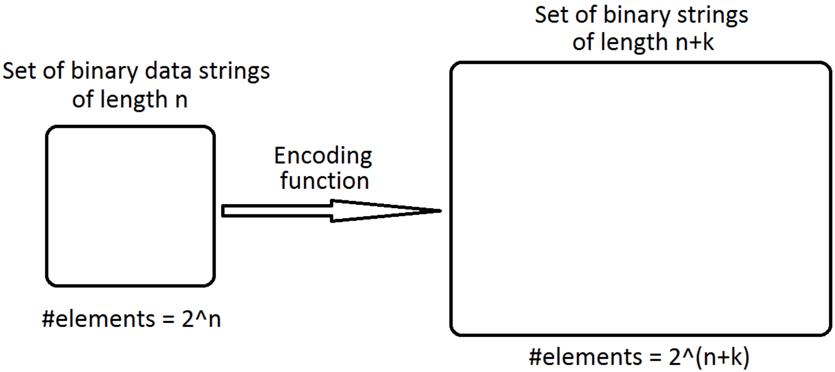
Concept block codes

For any two binary strings x and y:

- Hamming weight of x: The total number of 1's which appear in x.
- Hamming distance between x and y: The number of positions where x and y differ. (Same as Hamming weight of $(x \oplus y)$)

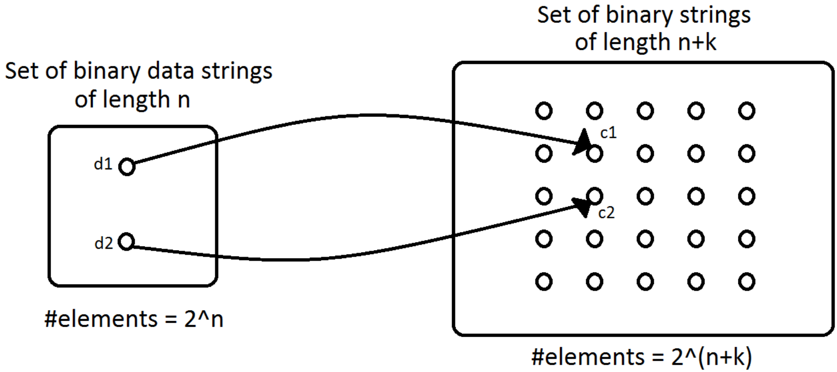
Concept block codes

For every data string, the encoder maps it to a longer binary string called a codeword.



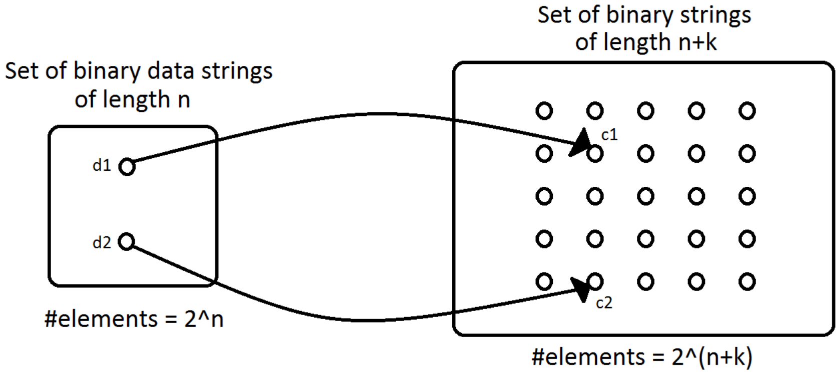
Concept block codes

Scenario 1: Two different data strings get mapped to two codewords which are close to each other.



Concept block codes

Scenario 2: Two different data strings get mapped to two codewords which are far from each other.



Concept block codes

Desired property of code

- The Hamming distance between all pairs of codewords is big.

Distance of a code: The minimum Hamming distance of all possible pairs of codewords.

If a code has distance d, then:

- can be used as a $d - 1$ error detecting code
- can be used as a $\lfloor \frac{d-1}{2} \rfloor$ error correcting code
- mix of the two above

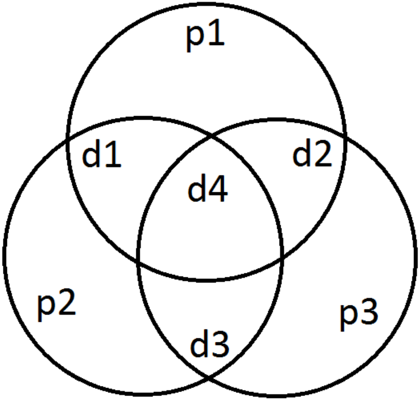
Hamming code (7,4)

1-bit error correcting linear block code.
4 data bits, 3 parity check bits.

Encoding function:
The data string $d_1 d_2 d_3 d_4$ gets encoded as $p_1 p_2 d_1 p_3 d_2 d_3 d_4$ with:

$$p_1 = d_1 \oplus d_2 \oplus d_4$$
$$p_2 = d_1 \oplus d_3 \oplus d_4$$
$$p_3 = d_2 \oplus d_3 \oplus d_4$$

Hamming code (7,4)



Hamming code (7,4)

See the binary strings as vectors from vectorspaces (with mod 2 addition).

$$\left(\mathbb{F}_2\right)^4 \to \left(\mathbb{F}_2\right)^7$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \mapsto \begin{pmatrix} x_1 + x_2 + x_4 \\ x_1 + x_3 + x_4 \\ x_1 \\ x_2 + x_3 + x_4 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Hamming code (7,4)

The encoding function f is obviously linear, since $\overline{a+b} = \overline{a} + \overline{b}$ with $\overline{x}$ defined as $x \bmod 2$.

Generator matrix for code f

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

Codeword for x is $c = Gx$.

Hamming code (7,4)

All codewords				
0000	0000000		1000	1110000
0001	1101001		1001	0011001
0010	0101010		1010	1011010
0011	1000011		1011	0110011
0100	1001100		1100	0111100
0101	0100101		1101	1010101
0110	1100110		1110	0010110
0111	0001111		1111	1111111

For any two codewords c_1, c_2, $(c_1 + c_2)$ is also a codeword:

$$c_1 + c_2 = Gx_1 + Gx_2 = G(x_1 + x_2)$$

So distance of Hamming code [7,4] is indeed 3.

Hamming code (7,4)

From all parity check equations, we get the parity check matrix

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

Note: For any codeword c, we have $Hc = 0$.

The decoder receives a binary string v, which can be written as $v = c + e$ with c the original codeword and e the error vector.

Hamming code (7,4)

How to decode v?

The syndrome of v is defined as $s = Hv$, but then we see:

$$s = Hv = H(c + e) = Hc + He = 0 + He = He$$

Answer: Solve $s = He$ for the error vector e which has the smallest Hamming weight.

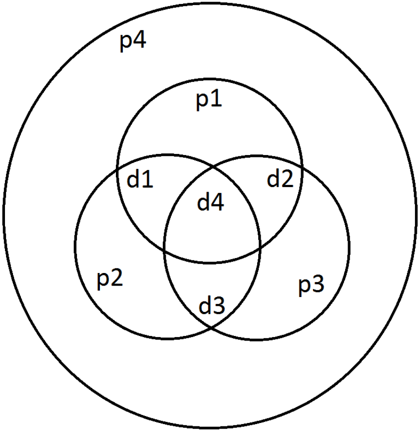
Hamming code (7,4)

Syndrome table	
000	0000000
100	1000000
010	0100000
110	0010000
001	0001000
101	0000100
011	0000010
111	0000001

Decoded codeword is $v - e = (c + e) - e = c$.

Extended Hamming code (8,4)

Use 1 extra parity bit to have 1-bit error correcting and 2-bit error detecting code.



Performance

Rate of a block code is (#data bits / blocksize).

Decoding error of source bit:
Repetition[3,1]:

$$p_b = \sum_{k=2}^{3} \binom{3}{k} \cdot f^k \cdot (1-f)^{(3-k)} \approx 3f^2$$

Hamming[7,4]:

$$p_B = \sum_{k=2}^{7} \binom{7}{k} \cdot f^k \cdot (1-f)^{(7-k)} \approx 21f^2$$

$$p_b \approx \frac{3}{7} \cdot p_B \approx 9f^2$$

Performance

With the chance of a bitflip occurring at $f = 0.01$:

	No ECC	Repitition[3,1]	Hamming[7,4]
Rate	1	$1/3$	$4/7$
Decoding error source bit	0.01	$3 \cdot 10^{-4}$	$9 \cdot 10^{-4}$

Performance

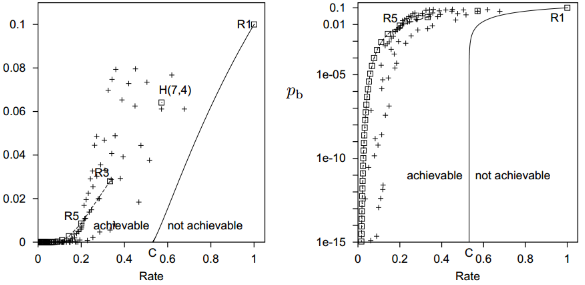


Figure 1.19. Shannon's noisy-channel coding theorem. The solid curve shows the Shannon limit on achievable values of (R, p_b) for the binary symmetric channel with $f = 0.1$. Rates up to $R = C$ are achievable with arbitrarily small p_b. The points show the performance of some textbook codes, as in figure 1.18.

The equation defining the Shannon limit (the solid curve) is $R = C/(1 - H_2(p_b))$, where C and H_2 are defined in equation (1.35).

$$C(f) = 1 - H_2(f) = 1 - \left[f \log_2 \frac{1}{f} + (1-f) \log_2 \frac{1}{1-f}\right]; \qquad (1.35)$$

Figuur: David J. C. MacKay. Information Theory, Inference, and Learning Algorithms.