TSKS02 Telecommunication

Lecture 10

Decoding of Error Control Codes

Benefits of Error Control Codes

Mikael Olofsson
Department of EE (ISY)
Div. of Communication Systems

LINKÖPINGS UNIVERSITET

Last time – Binary Linear Codes $[n,k,d]$

A vector space expressed in a basis

$$\mathscr{C} = \{\overline{m}G \;\forall\; \overline{m} \in \mathrm{GF}(2)^k\}$$

Generator matrix ($k \times n$), linearly independent rows.

$$\boxed{HG^{\mathrm{T}} = 0}$$

… the nullspace of a matrix

$$\mathscr{C} = \{\overline{c} \in \mathrm{GF}(2)^n : H\overline{c}^{\mathrm{T}} = \overline{0}\}$$

Parity check matrix ($(n-k) \times n$), linearly independent rows.

Length, n, # columns in G or H

Dimension, k, # rows in G.

<u>Minimum distance, d</u>

Smallest Hamming distance between different codewords.

Smallest Hamming weight of non-zero codewords.

Smallest number of linearly dependent columns in H.

Syndrome Decoding

Received vector: $\overline{x} = \overline{c} + \overline{e} = (x_1, \dots, x_n)$

Sent codeword $(c_1, \dots, c_n)$

Error vector $(e_1, \dots, e_n)$

We know: $H\overline{c}^{\mathrm{T}} = \overline{0}$

$\boxed{\text{Test that!}}$

Syndrome: $\overline{s} = H\overline{x}^{\mathrm{T}} = H \cdot (\overline{c} + \overline{e})^{\mathrm{T}} = \underbrace{H\overline{c}^{\mathrm{T}}}_{=\overline{0}} + H\overline{e}^{\mathrm{T}} = H\overline{e}^{\mathrm{T}}$

Notation: $H = \begin{pmatrix} \vdots & & \vdots \\ \overline{h}_1 & \cdots & \overline{h}_n \\ \vdots & & \vdots \end{pmatrix} \Rightarrow \overline{s} = \sum_{i=1}^{n} x_i \overline{h}_i = \sum_{i=1}^{n} e_i \overline{h}_i$

One error in position i. $\Rightarrow \overline{s} = \overline{h}_i$

Two errors in positions i & j. $\Rightarrow \overline{s} = \overline{h}_i + \overline{h}_j$

Find closest codeword.

$\Leftrightarrow$

Find smallest set of cols in H that sum up to the syndrome.

Binary Hamming Codes

Parameters:

$m = \#\text{rows in } H = n-k > 1$

$n = 2^m - 1$

$k = 2^m - m - 1$

$d = 3$

m	n	k
2	3	1
3	7	4
4	15	11
5	31	26
6	63	57
7	127	120
8	255	247
⋮	⋮	⋮

The parity check matrix H_m contains all non-zero m-dim vectors as columns.

Examples:

$$H_2 = \left(\underbrace{\begin{matrix}1&0\\0&1\end{matrix}}_{I_2}\;\vdots\;\underbrace{\begin{matrix}1\\1\end{matrix}}_{P_2^{\mathrm{T}}}\right) \Rightarrow G_2 = \left(\underbrace{1\;1}_{P_2}\;\vdots\;\underbrace{1}_{I_1}\right)$$

$$H_3 = \left(\underbrace{\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix}}_{I_3}\;\vdots\;\underbrace{\begin{matrix}1&1&0&1\\1&0&1&1\\0&1&1&1\end{matrix}}_{P_3^{\mathrm{T}}}\right) \Rightarrow G_3 = \left(\underbrace{\begin{matrix}1&1&0\\1&0&1\\0&1&1\\1&1&1\end{matrix}}_{P_3}\;\vdots\;\underbrace{\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix}}_{I_4}\right)$$

$$H_4 = \left(\underbrace{\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix}}_{I_4}\;\vdots\;\underbrace{\begin{matrix}1&1&1&0&0&0&1&1&1&0&1\\1&0&0&1&1&0&1&1&0&1&1\\0&1&0&1&0&1&1&0&1&1&1\\0&0&1&0&1&1&0&1&1&1&1\end{matrix}}_{P_4^{\mathrm{T}}}\right)$$

$$\Rightarrow \quad G_4 = (P_4 \;\vdots\; I_{11})$$

First Comparison 1(2)

Uncoded communication over BSC w. error probability p

$$n = k = 57, \quad d = 1$$

$$P_e = \mathbf{Pr}\{\text{at least one error among } k \text{ bits}\}$$
$$= 1 - \mathbf{Pr}\{\text{no errors among } k \text{ bits}\} = 1 - (1-p)^k \approx kp$$

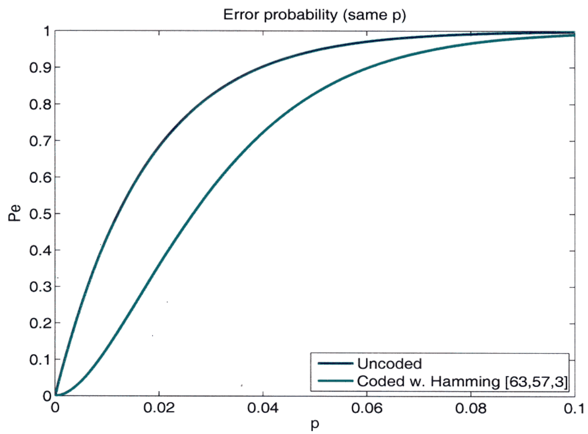
Coded communication over the same BSC

Encoding: Hamming [63,57,3]

$$k = 57, \quad n = 63, \quad d = 3$$

$$P_e = \mathbf{Pr}\{\text{at least two errors among } n \text{ bits}\}$$
$$= 1 - \mathbf{Pr}\{\text{zero or one errors among } n \text{ bits}\}$$
$$= 1 - (1-p)^n - np(1-p)^{n-1} \approx n(n-1)p^2/2$$

First Comparison 2(2)



Uncoded: $P_e = 1 - (1-p)^k$ Coded: $P_e = 1 - (1-p)^n - np(1-p)^{n-1}$

Second Comparison 1(2)

Compare with BSC with the same bit energy

Assume binary modulation and the same E_b in both cases.
That is, use signal energy $(k/n)E_b$ for each codeword bit.

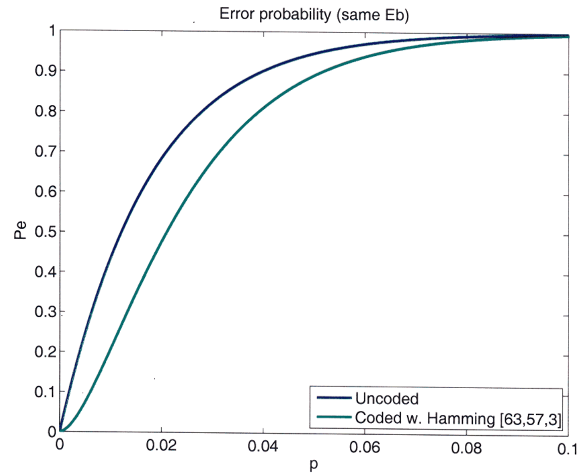
Result: BSC with error probability q, given by

$$q = Q\left(\sqrt{k/n} \cdot Q^{-1}(p)\right),$$

for the Hamming [63,57,3] code:

$$P_e = 1 - (1-q)^n - nq(1-q)^{n-1} \approx n(n-1)q^2/2$$

Second Comparison 2(2)



Dual Codes

$\mathscr{C}$: A binary linear code with generator matrix G and parity check matrix H.

$\mathscr{C}^\perp$: Its dual, a bin. linear code with gen. matrix H and parity check matrix G.

Note: $G^\perp = H, \quad H^\perp = G$ and $\mathscr{C}:[n,k] \Leftrightarrow \mathscr{C}^\perp:[n,n-k]$

Example: $\mathscr{C}$ is Hamming $[7,4,3]$

$$H = \begin{pmatrix} 0\,1\,1\,1 & 1\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0 \\ 1\,1\,0\,1 & 0\,0\,1 \end{pmatrix} = G^\perp$$

$$G = \begin{pmatrix} 1\,0\,0\,0 & 0\,1\,1 \\ 0\,1\,0\,0 & 1\,0\,1 \\ 0\,0\,1\,0 & 1\,1\,0 \\ 0\,0\,0\,1 & 1\,1\,1 \end{pmatrix} = H^\perp$$

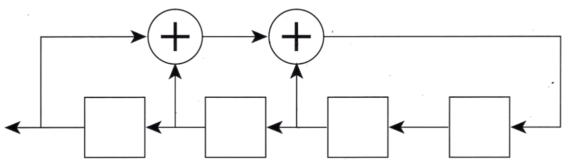
$\mathscr{C}^\perp$:

Info $\overline{m}$	Codeword $\overline{m}G^\perp$	Weight
000	0000000	0
001	1101001	4
010	1011010	4
011	0110011	4
100	0111100	4
101	1010101	4
110	1100110	4
111	0001111	4

$d^\perp = 4$

Cyclic [7,4,3] Hamming Code

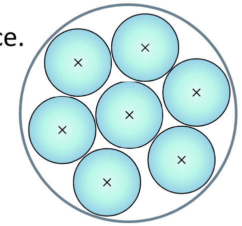
Cyclic code: A linear code where every cyclic shift of a codeword is a codeword.



c_1	c_2	c_3	c_4
c_2	c_3	c_4	c_5
c_3	c_4	c_5	c_6
c_4	c_5	c_6	c_7

The Hamming Bound

Based on packing spheres in the binary Hamming space.



A linear $[n,k,d]$ code:

Size of code: 2^k codewords.

Size of vector space: 2^n vectors.

Decoding sphere of radius $\lfloor (d-1)/2 \rfloor$ around each codeword.

Size of a sphere: $\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}$

Size of the union of spheres: $2^k \sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} \leq 2^n$

Disjoint spheres

Also called the sphere packing bound.

The Singleton Bound

Based on shortening codewords.

A linear $[n,k,d]$ code:

Remove the same $d-1$ coefficients in each codeword.

Result: A linear $[n',k',d']$ code with $n' = n-d+1$, $k' = k$, $d' \geq 1$.

Size of resulting code: $2^{k'} = 2^k$ codewords.

Size of resulting vector space: $2^{n'} = 2^{n-d+1}$ vectors.

Distinct codewords: $2^{n'} \geq 2^{k'} \Rightarrow 2^{n-d+1} \geq 2^k$

Distinct codewords

Result: $n-d+1 \geq k$

Usually written: $n-k \geq d-1$

The Maximum Value of the Minimum Distance of Binary Linear Block Codes

n \ k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	*1*																			
2	***2***	1																		
3	*3*	**2**	1																	
4	*4*	2	**2**	1																
5	*5*	3	2	**2**	1															
6	*6*	4	3	2	**2**	1														
7	*7*	4	4	3	2	**2**	1													
8	*8*	5	4	4	2	2	**2**	1												
9	*9*	6	4	4	3	2	2	**2**	1											
10	*10*	6	5	4	4	3	2	2	**2**	1										
11	*11*	7	6	5	4	4	3	2	2	**2**	1									
12	*12*	8	6	6	4	4	4	3	2	2	**2**	1								
13	*13*	8	7	6	5	4	4	4	3	2	2	**2**	1							
14	*14*	9	8	7	6	5	4	4	4	3	2	2	**2**	1						
15	*15*	10	8	8	7	6	5	4	4	4	3	2	2	**2**	1					
16	*16*	10	8	8	8	6	6	5	4	4	4	2	2	2	**2**	1				
17	*17*	11	9	8	8	7	6	6	5	4	4	3	2	2	2	**2**	1			
18	*18*	12	10	8	8	8	7	6	6	4	4	4	3	2	2	2	**2**	1		
19	*19*	12	10	9	8	8	8	7	6	5	4	4	4	3	2	2	2	**2**	1	
20	*20*	13	11	10	9	8	8	8	7	6	5	4	4	4	3	2	2	2	**2**	1

Data fetched from http://www.win.tue.nl/~aeb/voorlincod.html
on March 14, 2005.

Italic Repetition codes
Bold Parity check codes
Hamming codes
Simplex codes
Code used in text TV

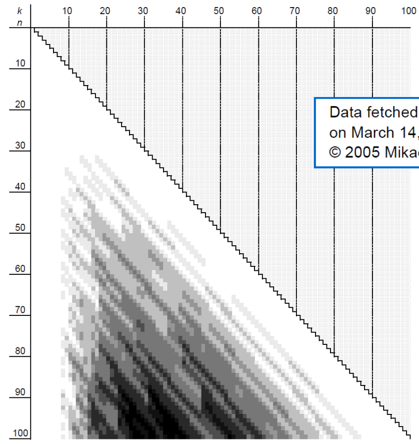
Upper and Lower Bounds on the Maximum Value of the Minimum Distance of Binary Linear Block Codes

n \ k	10	11	12	13	14	15	16	17	18	19	20
30	11	10	9	8	8	8	7	6	6	6	5
31	12	11	10	9	8	8	8	7	6	6	6
32	12	12	10	10	8-9	8	8	8	6-7	6	6
33	12	12	11	10	9-10	8-9	8	8	7-8	6-7	6
34	12	12	12	10	10	9-10	8-9	8	8	7-8	6-7
35	12-13	12	12	11	10	10	9-10	8	8	8	7-8
36	13-14	12-13	12	12	11	10	10	8-9	8	8	8
37	14	13-14	12-13	12	12	10-11	10	9-10	8-9	8	8
38	14	14	13-14	12	12	11-12	10-11	10	9-10	8-9	8
39	15	14	14	12-13	12	12	11-12	10-11	10	9-10	8-9
40	16	14-15	14	12-14	12-13	12	12	11-12	10-11	10	9-10

Data fetched from http://www.win.tue.nl/~aeb/voorlincod.html
on March 14, 2005.

No difference
Difference is 1
Difference is 2

Difference Between Upper and Lower Bounds for the Maximum Value of the Minimum Distance of Binary Linear Block Codes



Data fetched from http://www.win.tue.nl/~aeb/voorlincod.html
on March 14, 2005.

Difference is 0 1 2 3 4 5 6 7 8

Integer and Polynomial Division

Integer division

Ex: $\frac{1732}{15} = 115 + \frac{7}{15}$

```
     115 ← Quotient
15 | 1732
    -15
      23
     -15
       82
      -75
        7 ← Remainder
```

Polynomial division (binary polynomials)

Ex: $\frac{x^5+x^3+x+1}{x^2+x+1} = x^3+x^2+x+\frac{1}{x^2+x+1}$

$$
\begin{array}{l}
1\cdot x^3+1\cdot x^2+1\cdot x+0\cdot 1 \\
x^2+x+1 \;\big|\; 1\cdot x^5+0\cdot x^4+1\cdot x^3+0\cdot x^2+1\cdot x+1\cdot 1 \\
1\cdot x^5+1\cdot x^4+1\cdot x^3 \\
\hline
1\cdot x^4+0\cdot x^3+0\cdot x^2 \\
1\cdot x^4+1\cdot x^3+1\cdot x^2 \\
\hline
1\cdot x^3+1\cdot x^2+1\cdot x \\
1\cdot x^3+1\cdot x^2+1\cdot x \\
\hline
0\cdot x^2+0\cdot x+1\cdot 1 \\
0\cdot x^2+0\cdot x+0\cdot 1 \\
\hline
0\cdot x+1\cdot 1
\end{array}
$$

With bits only:

```
       1110
111 | 101011
      111
       100
       111
        111
        111
         001
         000
          01
```

CRC Codes – Division Algorithms

CRC = Cyclic Redundancy Check

Division Algorithm for Integers (over 2000 years old wisdom) :

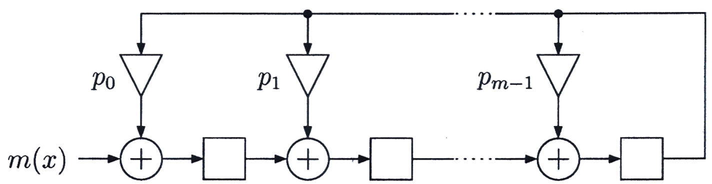
Given integers a and b, $b \neq 0$. Then there exist unique integers q and r, $0 \leq r < |b|$, such that $a = qb + r$ holds.

Division Algorithm for Binary Polynomials (slightly newer wisdom):

Given binary polynomials $a(x)$ and $b(x)$, $b(x) \neq 0$. Then there exist unique binary polynomials $q(x)$ and $r(x)$, $\deg\{r(x)\} < \deg\{b(x)\}$, such that $a(x) = q(x)b(x) + r(x)$ holds.

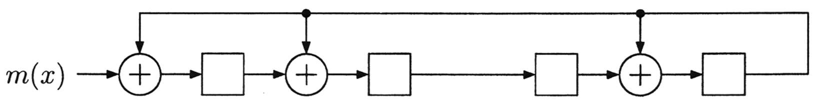
CRC Codeword Generation

$$p(x) = \sum_{i=0}^{m} p_i x^i, \qquad p_m = 1, \qquad m = n - k.$$



Example:

$p(x)$: $1 + 1 \cdot x + 0 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4$



Mikael Olofsson

ISY/CommSys

www.liu.se

LINKÖPINGS UNIVERSITET