Theory and Design of Turbo and Related Codes

Lecture 2

Jossy Sayir & Gottfried Lechner

http://userver.ftw.at/~jossy/turbo/index.htm

Kplus Kompetenzzentren-Programm

What we have learned last time...

Linear codes

A code is a K-dimensional subspace of $GF(2)^N$

$$(x_1\ x_2 \ldots\ x_{16}) = (u_1 \ldots\ u_6)\begin{pmatrix} 0&1&1&0&0&0&0&0&1&0&1&1&1&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&0&1&0&1&1&0 \\ 1&1&0&0&0&0&0&1&0&1&0&0&0&0&1&1 \\ 0&0&0&0&1&1&1&0&1&1&1&0&1&1&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0 \\ 0&0&1&1&0&0&0&1&0&0&1&1&0&1&0&0 \end{pmatrix}$$

Codeword ↑ Source Sequence ↑ Encoder Matrix ↑

➔ Information Theory for the Binary Erasure Channel

What we will do today...

Systematic Codes

Parity-Check Matrices

Low-Density Parity-Check (LDPC) Codes
Definitions, constructions, decoding and performance for the Binary Erasure Channel

Systematic Encoding

Our encoding matrix from the first lecture:

$$G = \left(\begin{array}{cccc|cccc|cccc|cccc}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
\hline
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0
\end{array}\right)$$

Any linear row operations leave the code unchanged but give us another encoder

Systematic Encoding (continued)

$$G' = \left(\begin{array}{cccccc|cccccccccc} 1&1&0&0&0&0&0&1&0&1&0&0&0&0&1&1 \\ 0&1&1&0&0&0&0&0&1&0&1&1&1&0&0&0 \\ 0&0&1&1&0&0&0&1&0&0&1&1&0&1&0&0 \\ \hline 0&0&0&1&1&0&1&0&0&0&0&1&0&1&1&0 \\ 0&0&0&0&1&1&1&0&1&1&1&0&1&1&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0 \end{array}\right) \begin{array}{l} g_1 \\ g_2 \\ g_6 \\ g_3 \\ g_4 \\ g_5 \end{array}$$

Systematic encoding matrix:

$$G'' = \left(\begin{array}{cccccc|cccccccccc} 1&0&0&0&0&0&1&1&0&0&1&1&1&1&0&1 \\ 0&1&0&0&0&0&1&0&0&1&1&1&1&1&1&0 \\ 0&0&1&0&0&0&1&0&1&1&0&0&0&1&1&0 \\ \hline 0&0&0&1&0&0&1&1&1&1&1&1&0&0&1&0 \\ 0&0&0&0&1&0&0&1&1&1&1&0&0&1&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0 \end{array}\right) \begin{array}{l} g_1+g_2+g_3+g_4+g_5+g_6 \\ g_2+g_3+g_4+g_5+g_6 \\ g_3+g_4+g_5+g_6 \\ g_4+g_5+g_6 \\ g_5+g_6 \\ \end{array}$$

Systematic Encoding (continued)

Systematic Encoder:

$$(x_1 \ldots x_{16}) = (u_1 \ldots u_6)\, G'' = (\underbrace{u_1 \ldots u_6}_{\text{Systematic part}}\ \overbrace{x_7 \ldots x_{16}}^{\text{Parity-check part}})$$

For example: u = (111111) , x = (111111 1010011011)

The encoder matrix must be of the form: $G = [I\ P]$

Parity-Check Matrix

Systematic Parity-Check Matrix: $H = [P^T_{M \times K} \; I_{M \times M}]$

$$
\begin{aligned}
G H^T &= [I_{K \times K} \; P_{K \times M}] \, [P^T_{M \times K} \; I_{M \times M}]^T \\
&= I_{K \times K} \, P_{K \times M} + P_{K \times M} \, I_{M \times M} \\
&= P_{K \times M} + P_{K \times M} \\
&= 0
\end{aligned}
$$

For any codeword $v \in V$, we can write $v = u\,G$, therefore

$$v\,H^T = u\,G\,H^T = 0$$

Furthermore, any sequence x such that $x\,H^T = 0$ is a codeword.

Parity-Check Matrix (continued)

Systematic encoder matrix:

$$
G_{sys} = \left(\begin{array}{cccccc:cccccccccc}
1&0&0&0&0&0&1&1&0&0&1&1&1&1&0&1\\
0&1&0&0&0&0&1&0&0&1&1&1&1&1&1&0\\
0&0&1&0&0&0&1&0&1&1&0&0&0&1&1&0\\
0&0&0&1&0&0&1&1&1&1&1&1&0&0&1&0\\
0&0&0&0&1&0&0&1&1&1&1&0&0&1&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0
\end{array}\right)
$$

Systematic parity-check matrix:

$$
H_{sys} = \left(\begin{array}{cccccc:cccccccccc}
1&1&1&1&0&1&1&0&0&0&0&0&0&0&0&0\\
1&0&0&1&1&1&0&1&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&1&1&1&0&0&0&0&1&0&0&0&0&0&0\\
1&1&0&1&1&0&0&0&0&0&1&0&0&0&0&0\\
1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
1&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
1&1&1&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right)
$$

Parity-Check Matrix (continued)

$$[1\ 0\ 1\ 0\ 0\ 1] \begin{pmatrix} 1&0&0&0&0&0&1&1&0&0&1&1&1&1&0&1 \\ 0&1&0&0&0&0&1&0&0&1&1&1&1&1&1&0 \\ 0&0&1&0&0&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&0&1&0&0&1&1&1&1&1&1&0&0&1&0 \\ 0&0&0&0&1&0&0&1&1&1&1&0&0&1&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0 \end{pmatrix} = [1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1]$$

$(u_1 \ldots u_6)$ G = $(x_1 \ldots x_{16})$

$$[1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1] \begin{pmatrix} 1&1&0&0&1&1&1&1&0&1 \\ 1&0&0&1&1&1&1&1&1&0 \\ 1&0&1&1&0&0&0&1&1&0 \\ 1&1&1&1&1&1&0&0&1&0 \\ 0&1&1&1&1&0&0&1&0&0 \\ 1&1&0&0&0&0&1&0&0&0 \\ \hline 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} =$$

$(x_1 \ldots x_{16})$ H^T =

Parity-Check Matrix (continued)

The parity-check matrix is an encoder matrix of the dual code.

⇒ H is an $(N-K)\times N$ matrix
(G is a $K\times N$ matrix)

Again, any linear row operations leave the (dual) code unchanged and yield another valid parity-check matrix for the same code.

Decoding with Parity-Check Matrices

Received: (1 x x x x 1 0 0 0 1 x 1 x x x 1)

$$H_{sys} = \left(\begin{array}{cccccc|cccccccccc}
1&1&1&1&0&1&1&0&0&0&0&0&0&0&0&0\\
1&0&0&1&1&1&0&1&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&1&1&1&0&0&0&0&1&0&0&0&0&0&0\\
1&1&0&1&1&0&0&0&0&0&1&0&0&0&0&0\\
1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
1&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
1&1&1&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right)$$

Permutation:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	A	C	D	E	0	5	6	7	8	9	B	F

Decoding with Parity-Check Matrices

Received: (x x x x x x x x 1 1 0 0 0 1 1 1)

$$H = \left(\begin{array}{cccccccccccccccc}
1&1&1&0&0&0&0&0&1&1&1&0&0&0&0&0\\
0&0&1&1&0&0&0&0&1&1&0&1&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&1&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&0\\
1&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0\\
1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0\\
1&1&0&1&0&0&1&0&1&0&0&0&0&0&0&0\\
1&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1
\end{array}\right)$$

Triangulate…

Decoding with Parity-Check Matrices

Received: (x x x x x x x x 1 1 0 0 0 1 1 1)

$$
H = \left(\begin{array}{cccccccccccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0
\end{array}\right)
$$

Solve erasures…

Decoding with Parity-Check Matrices

(1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 1)

$$
H = \left(\begin{array}{cccccccccccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0
\end{array}\right)
$$

Invert permutation… Cut out systematic part…

Decoding with Parity-Check Matrices

Permutation:

(1 1 0 1 1 1 ⁞ 0 0 0 1 0 1 1 1 0 1)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	A	C	D	E	0	5	6	7	8	9	B	F

(1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 1)

Decoded: [1 1 0 1 1 1]

Trick question: did we obtain the same solution as in the first lecture?

Another Parity-Check Matrix...

Received: (1 X X X X 1 0 0 0 1 X 1 X X X 1)

Parity-Check Matrix:

$$\left(\begin{array}{cccccc|cccccccccc}
0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&1\\
0&0&0&1&0&1&1&0&0&1&0&0&0&0&0&0\\
0&0&1&0&1&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&1&1&0&0\\
1&0&0&1&0&0&0&1&0&0&1&0&0&0&0&0
\end{array}\right)$$

Iterative Decoding

Step 1

(1 X X 0 X 1 0 0 0 1 X 1 X X X 1)

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

Iterative Decoding

Step 2

(1 X X 0 X 1 0 0 0 1 X 1 X 1 X 1)

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

Iterative Decoding

ftw.

Step 3

(1 X X 0 X 1 0 0 0 1 1 1 X 1 X 1)

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

ftw.

Step 4

(1 X X 0 X 1 0 0 0 1 1 1 X 1 0 1)

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

Step 5

(1 X 1 0 X 1 0 0 0 1 1 1 X 1 0 1)

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

Step 6

(1 1 1 0 X 1 0 0 0 1 1 1 X 1 0 1)

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

Step 7

(1 1 1 0 **0** 1 0 0 0 1 1 1 X 1 0 1)

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

Decoded: (1 1 1 0 X 1 0 0 0 1 1 1 **1** 1 0 1)

Step 8

0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0

Iterative Decoding

We were lucky. This will not always work…

What property must a parity-check matrix have for iterative decoding to succeed with a high probability?

In effect, iterative decoding is possible in this setting whenever the matrix can be triangulated by simple row & column swaps

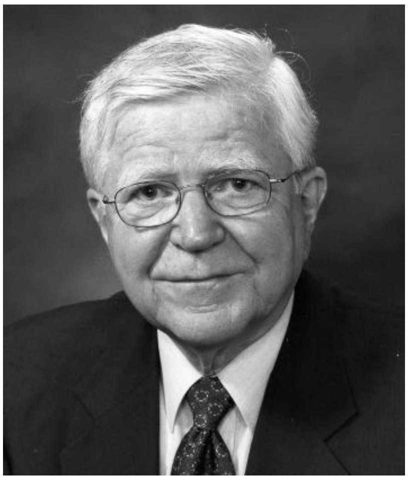
Low-Density Parity-Check Matrix

																Row weights
0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	4
0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	4
0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	4
1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	3
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	4
0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	4
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	3
0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	3
0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	4
1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	4
3	2	2	3	2	2	2	2	2	2	3	2	2	3	2	2	Column weights

The parity-check matrix must have a low density.

Low-Density Parity-Check Coding (Slide from 1st lecture)

Invented by Robert G. Gallager in his PhD thesis, MIT, 1963



From the Website of the Eduard Rhein Stiftung

Bob Gallager

(re-discovered by MacKay from Cambridge, England in 1999)

Regular Codes

																				Row weights
0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	6
0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	6
1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	1	0	6
0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1	6
1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	6
0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	6
0	0	0	0	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	6
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	6
0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	6
0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	6

3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

Column weights $d_v = 3$

Row weights $d_c = 6$

Columns and rows in the parity-check matrix of a regular LDPC code have fixed weights d_v and d_c (as opposed to irregular LDPC codes)

Regular Codes

How many '1's are there in the parity-check matrix?

Answer: $n_1 = d_v N = 3\times 20 = 60$
$= d_c (N-K)$

The design rate of the code is: $R = K/N = 1 - d_v/d_c$

If the rows of the parity-check matrix are linearly independent, the design rate is the true code rate

Code construction

We can specify the class of LDPC codes by (d_v, d_c) and the block length N.

Example: the matrix 2 pages ago was a (3,6) regular low-density parity-check matrix for a code of length N = 20

Warning: the matrix or the code are not uniquely determined by these specifications…

→ Random & deterministic code construction…

Example Deterministic Construction

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

α is a $q \times q$ unit matrix
q must be prime
$q \geq d_c$

$$H = \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 \end{pmatrix}$$

$\alpha^0 = I$ (identity matrix)
$\alpha^j = \alpha^{j \bmod q}$

Proposed by Evangelos Eleftheriou & Al. from IBM Research in Rüschlikon, Switzerland…

Intuitive Effect of d_v and d_c

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

Row weights $d_c = 6$

Column weights $d_v = 3$

A large d_v is better for the variables concerned!

A large d_c is worse for the p.c. equation concerned!

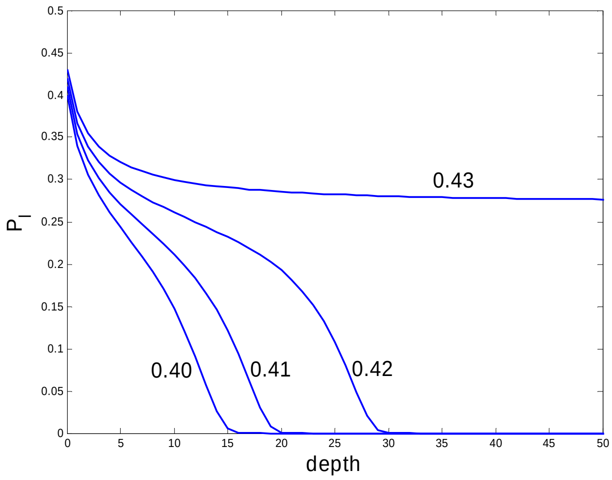
Density Evolution for the Erasure Channel:

$$p_{k+1} = p_0(1 - (1 - p_k)^{d_c - 1})^{d_v - 1}$$

Applying the recursion

$$p_{k+1} = p_0(1-(1-p_k)^{d_c-1})^{d_v-1}$$

d_v=3 d_c=6



Threshold:
0.42944