TSKS01 Digital Communication
Lecture 10

Bounds on good codes, convolutional codes

Emil Björnson

Department of Electrical Engineering (ISY)

Division of Communication Systems

COMMUNICATION SYSTEMS

LINKÖPING UNIVERSITY

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

A vector space expressed in a basis

$$\mathcal{C} = \{\bar{m}G \;\; \forall \bar{m} \in \mathbb{F}_2^k\}$$

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

… the nullspace of a matrix

$$\mathcal{C} = \{\bar{c} \in \mathbb{F}_2^n : H\bar{c}^T = \bar{0}\}$$

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

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

Length: n, # columns in G or H

Dimension: k, # rows in G.

Minimum distance, d

Smallest Hamming distance between different codewords.

Smallest Hamming weight of non-zero codewords.

Smallest number of linearly dependent columns in H.

Outline of This Lecture

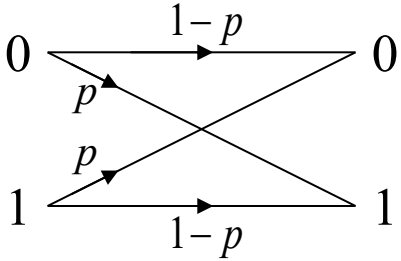
- Block codes
  - Benefits of error control codes
  - Soft decoding
  - Bounds on good codes
- Convolutional codes
  - Introduction

Error Probability: First comparison 1(2)

Uncoded communication over BSC with error probability p

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

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



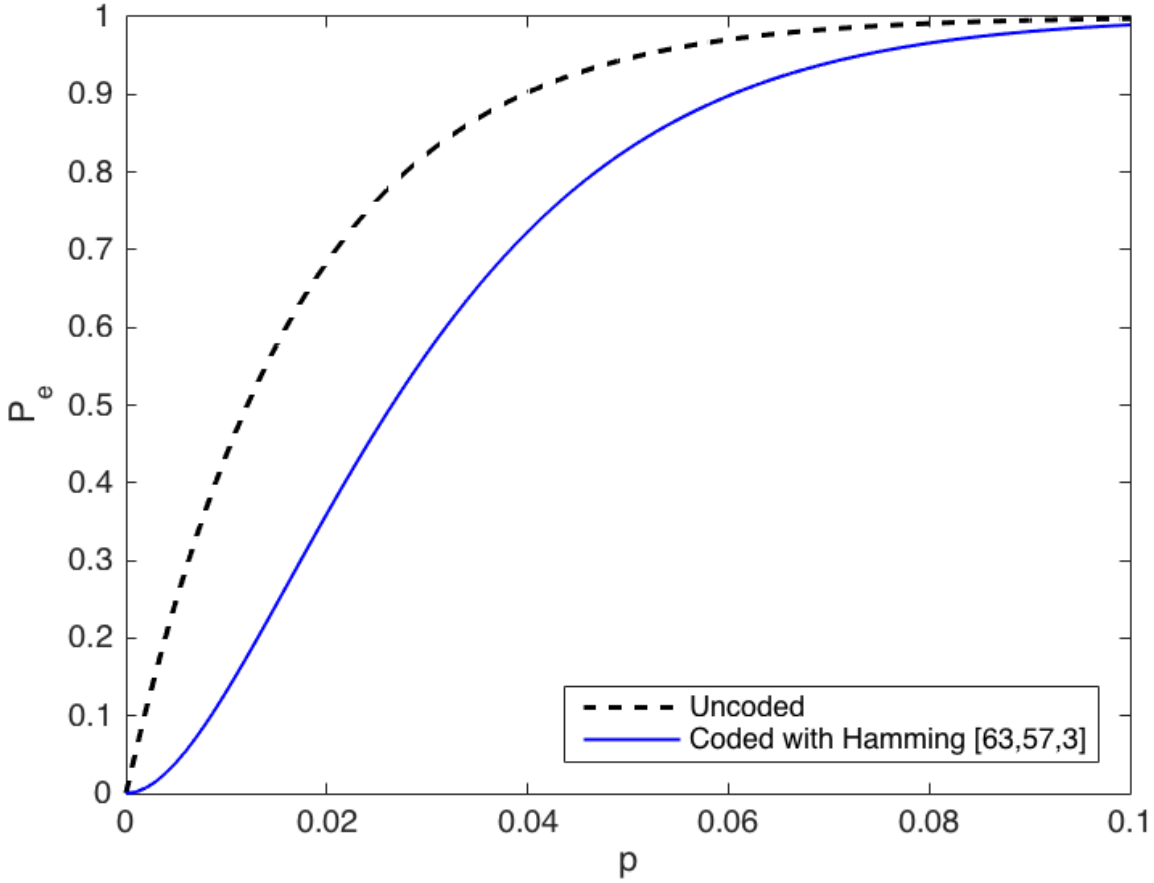
Coded communication over the same BSC

Encoding: Hamming [63,57,3]

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

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

Error Probability: First comparison 2(2)



Uncoded: $P_e = 1 - (1-p)^k$

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

Error Probability: Second comparison 1(2)

Unfair comparison if the total signal energy is different!

Uncoded and coded communication with same bit energy

Assume binary modulation and the same E_b in both cases:

- Use signal energy $(k/n)E_b$ for each codeword bit

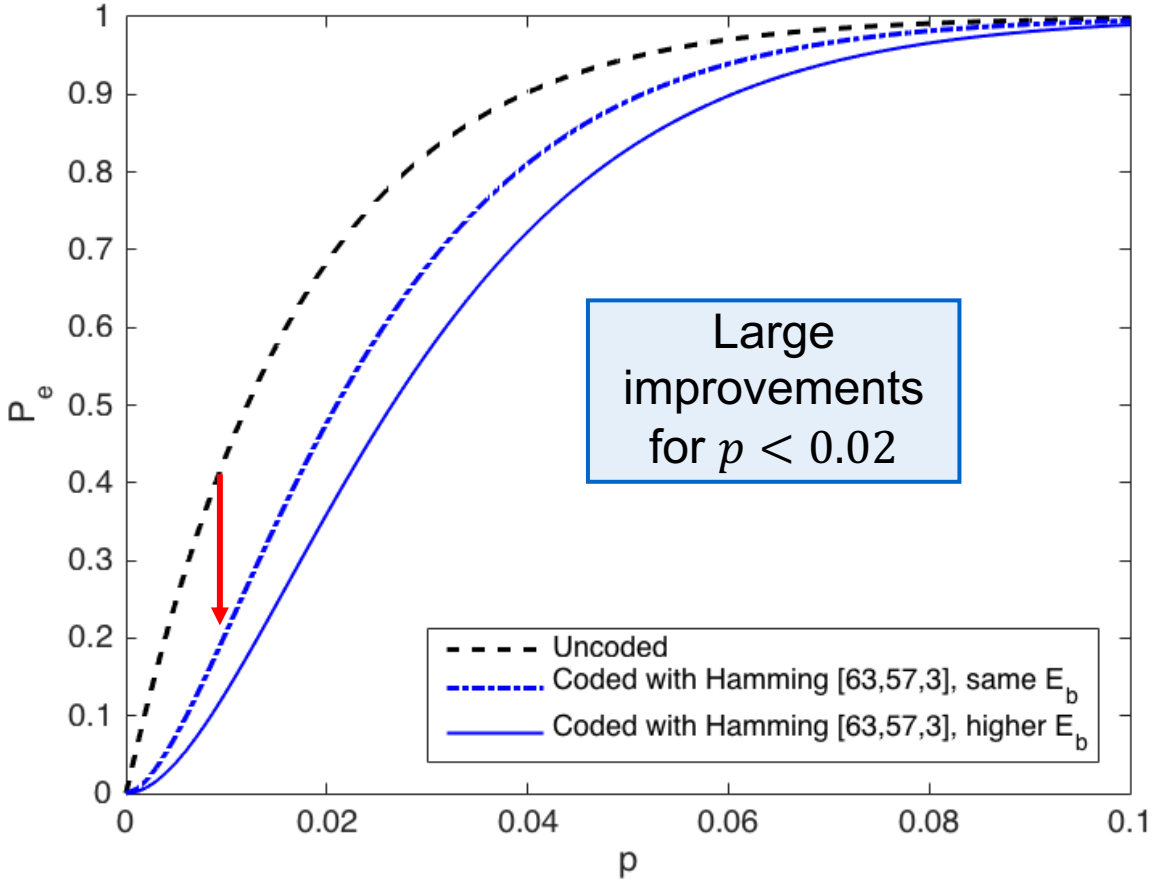
Result: BSC with error probability $q = Q\left(\sqrt{\frac{k}{n}}Q^{-1}(p)\right)$

Note:
$n \geq k \quad \rightarrow \quad q \geq p$

Uncoded: $P_e = 1 - (1-p)^k$

Hamming [63,57,3] code: $P_e = 1 - (1-q)^n - nq(1-q)^{n-1}$

Error Probability: Second comparison 2(2)



Uncoded: $P_e = 1 - (1-p)^k$

Coded: $P_e = 1 - (1-q)^n - nq(1-q)^{n-1}$

Soft Detection

- Recall: ML decision rules

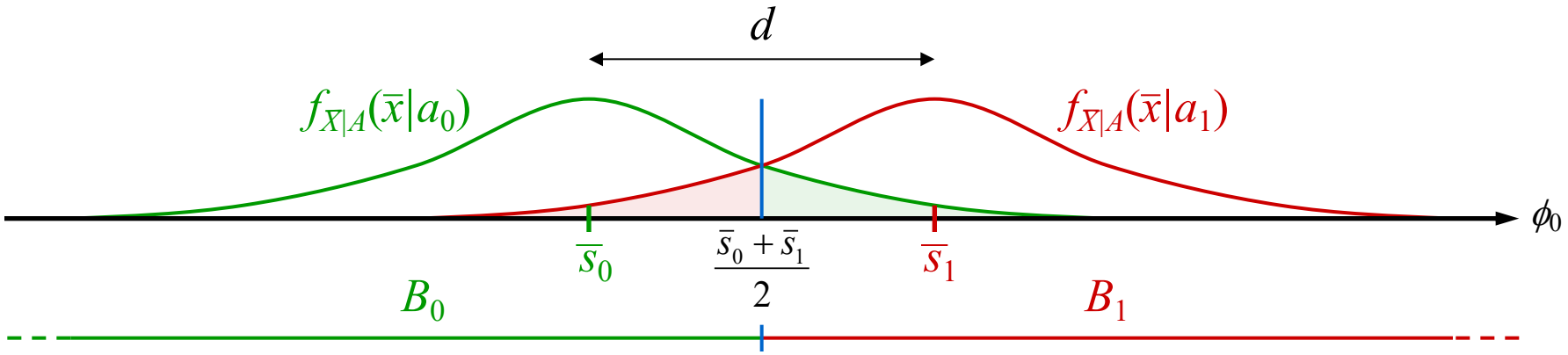
ML decision rule: Set $\hat{a} = a_i$ if $f_{\bar{X}|A}(\bar{x}|a_k)$ is maximized for $k = i$.

Equivalent ML rule: Set $\hat{a} = a_i$ if $d(\bar{x}, \bar{s}_k)$ is minimized for $k = i$.

- These rules give mechanisms to make decisions
  - How certain are we that are our decisions are correct?

  - **Hard decision**: $\hat{a} = a_i$
  - **Soft decision**: $\hat{a} = a_i$ and likelihood of right decision

Soft Detection: Likelihood of Being Right



ML rule: Select $\hat{a} = a_1$ if $f_{\bar{X}|A}(\bar{x}|a_1) > f_{\bar{X}|A}(\bar{x}|a_0)$

- Uncertain if $f_{\bar{X}|A}(\bar{x}|a_1) \approx f_{\bar{X}|A}(\bar{x}|a_0)$
- Very certain if $f_{\bar{X}|A}(\bar{x}|a_1) \gg f_{\bar{X}|A}(\bar{x}|a_0)$

Soft Detection: Log-Likelihood Ratio

- Measure of certainty:

$$\text{Is } \frac{f_{\bar{X}|A}(\bar{x}|a_1)}{f_{\bar{X}|A}(\bar{x}|a_0)} \approx 1 \text{ or } \frac{f_{\bar{X}|A}(\bar{x}|a_1)}{f_{\bar{X}|A}(\bar{x}|a_0)} \gg 1?$$

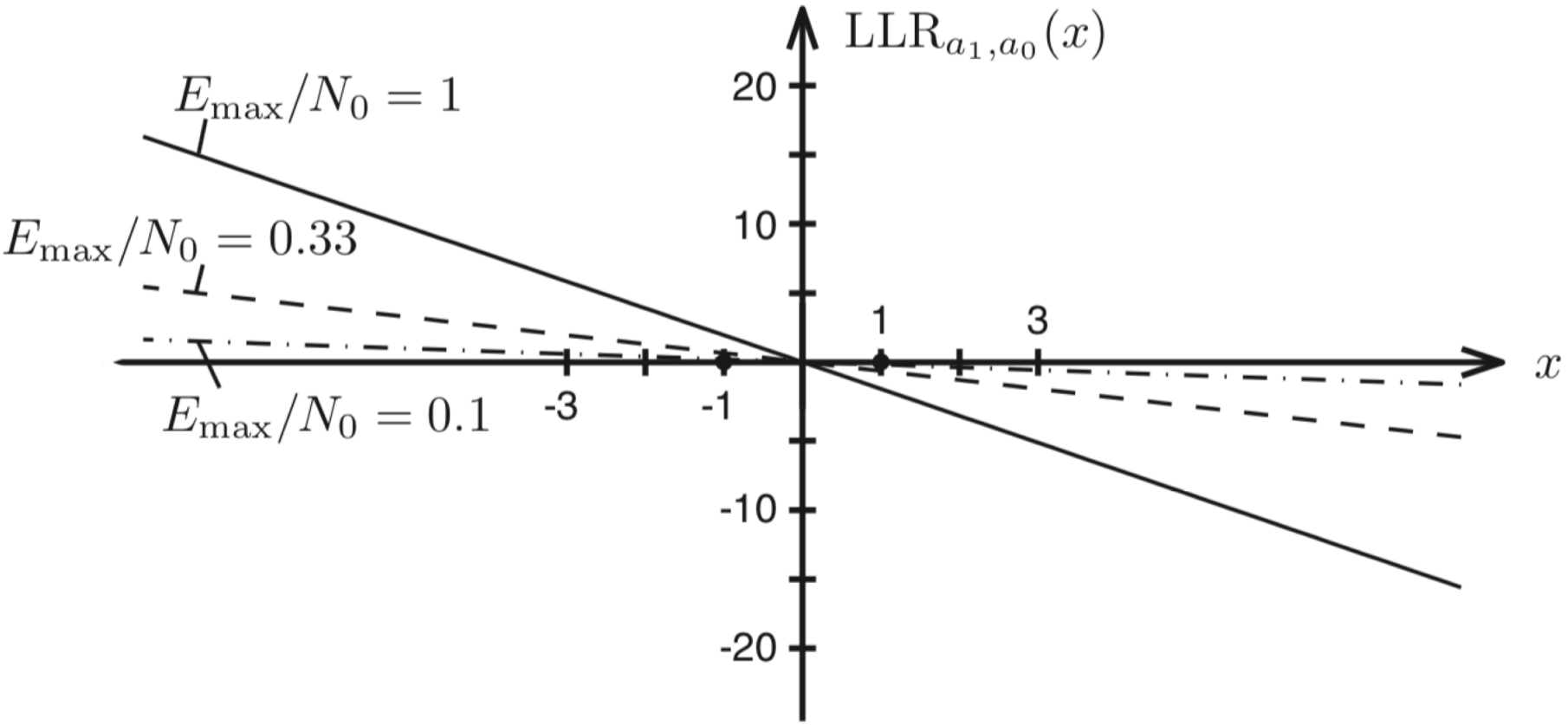
Definition: Log-likelihood ratio (LLR)

$$\mathrm{LLR}_{a_1,a_0}(\bar{x}) = \log\left(\frac{f_{\bar{X}|A}(\bar{x}|a_1)}{f_{\bar{X}|A}(\bar{x}|a_0)}\right)$$

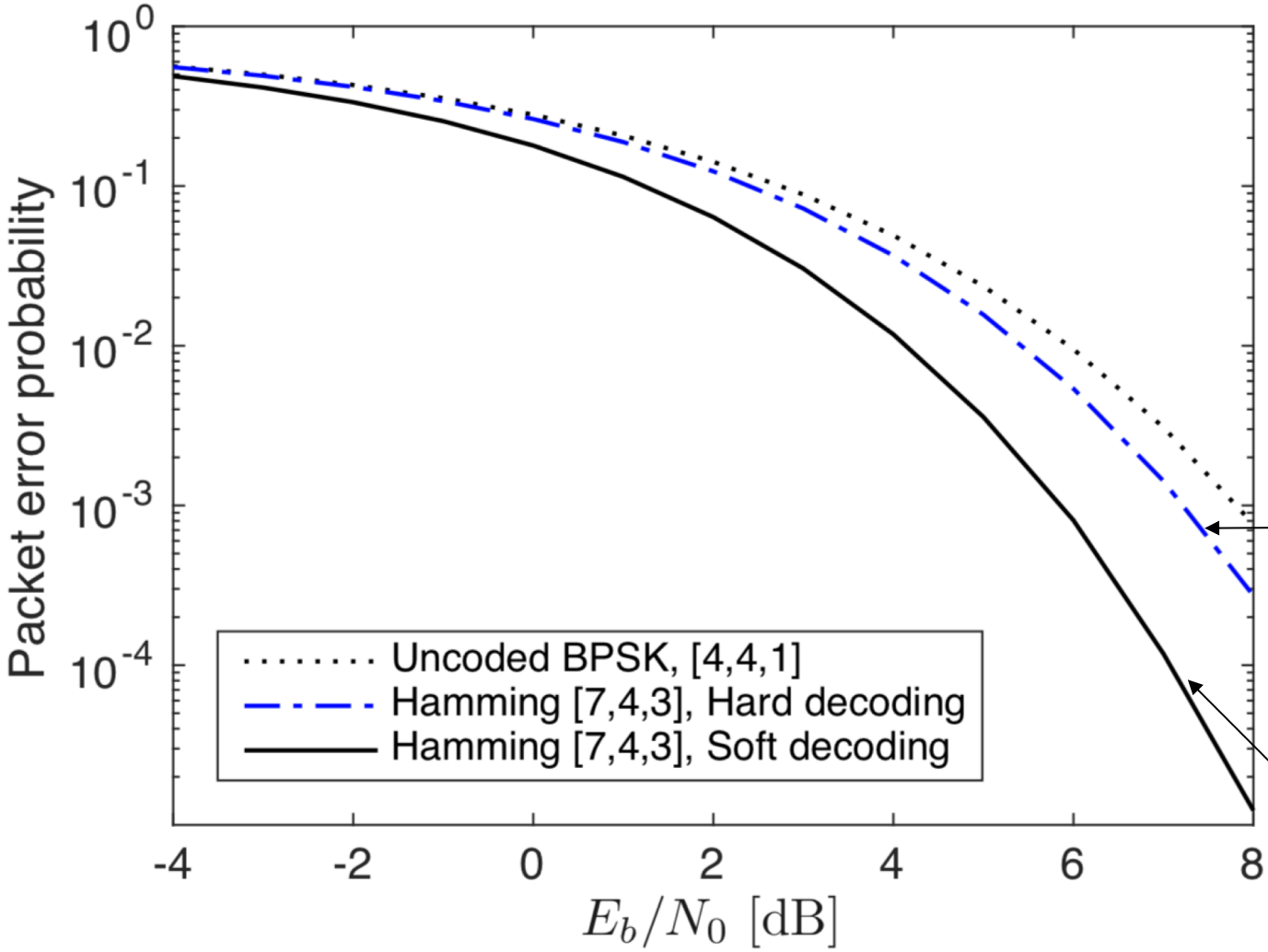
- $\mathrm{LLR}_{a_1,a_0}(\bar{x}) \approx 0$: Uncertain
- $\mathrm{LLR}_{a_1,a_0}(\bar{x}) \gg 0$: Certain that $\hat{a} = a_1$
- $\mathrm{LLR}_{a_1,a_0}(\bar{x}) \ll 0$: Certain that $\hat{a} = a_0$

Useful information to evaluate end result

Example: Log-Likelihood Ratio



Example: Soft and Hard Decoding of Hamming Code Over AWGN channel



Achieved by syndrome decoding

Achieved by maximizing LLRs or minimizing $\bar{x}\bar{c}^T$

What is a Good Block Code?

There only exist $[n, k, d]$ codes for some n, k, and d

Natural questions:

- Given n and k, what is the largest minimal distance d?
- Given n and d, how many bits k can be sent?
- Given k and d, what is the smallest possible n?

Hard to answer exactly!

Mathematical bounds: Shows what does *not* exist

Computer simulation: Shows what does exit

The Hamming Bound:

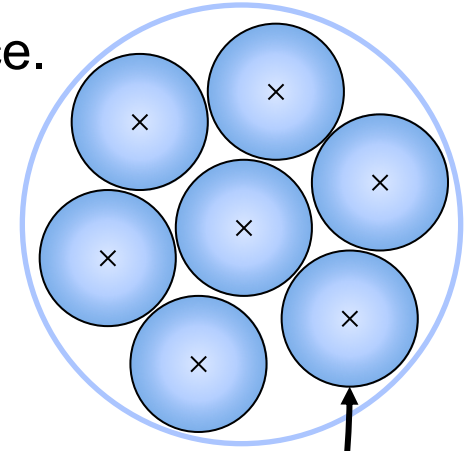
Based on packing spheres in the binary Hamming space.

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.



Disjoint spheres

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

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

Definition: *Perfect codes* achieves Hamming bound

(All received signals can be corrected)

Perfect codes:

Hamming codes

Repetition codes with odd n

The Singleton Bound

Based on shortening of 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: $2^{n'} \geq 2^{k'} \Rightarrow 2^{n-d+1} \geq 2^k$

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

Distinct codewords

Singleton bound: $n - k \geq d - 1$

(Shows which $[n, k, d]$ might be possible)

Equality achieved by:

Repetition codes

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

k / n	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

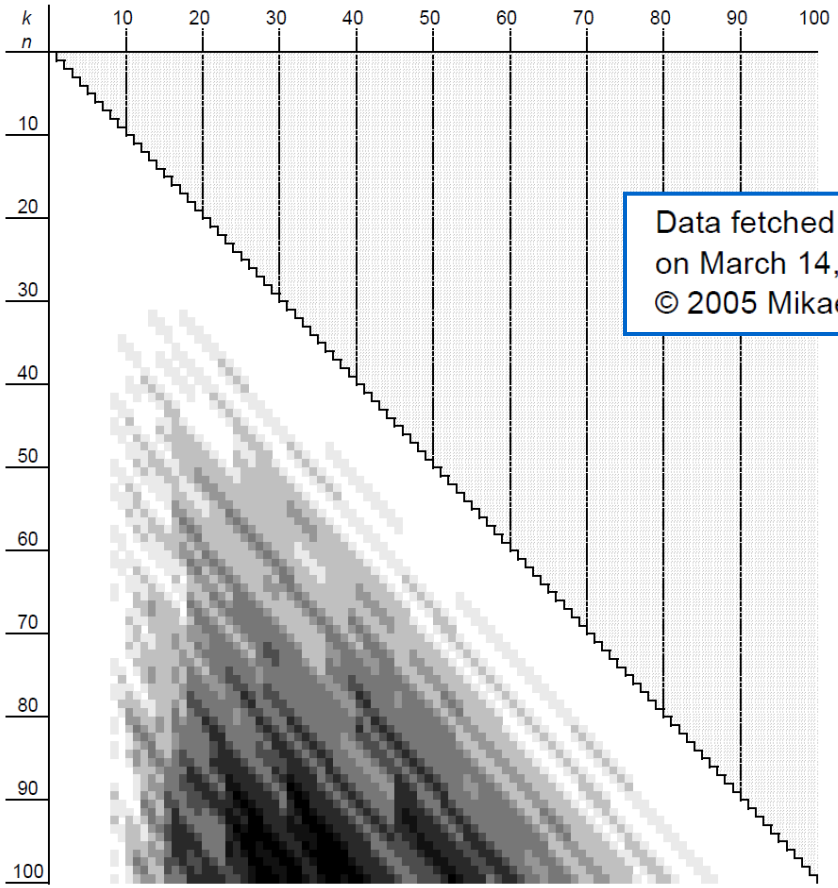
Upper and Lower Bounds on the Maximum Value of the Minimum Distance of Binary Linear Block Codes

k / n	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

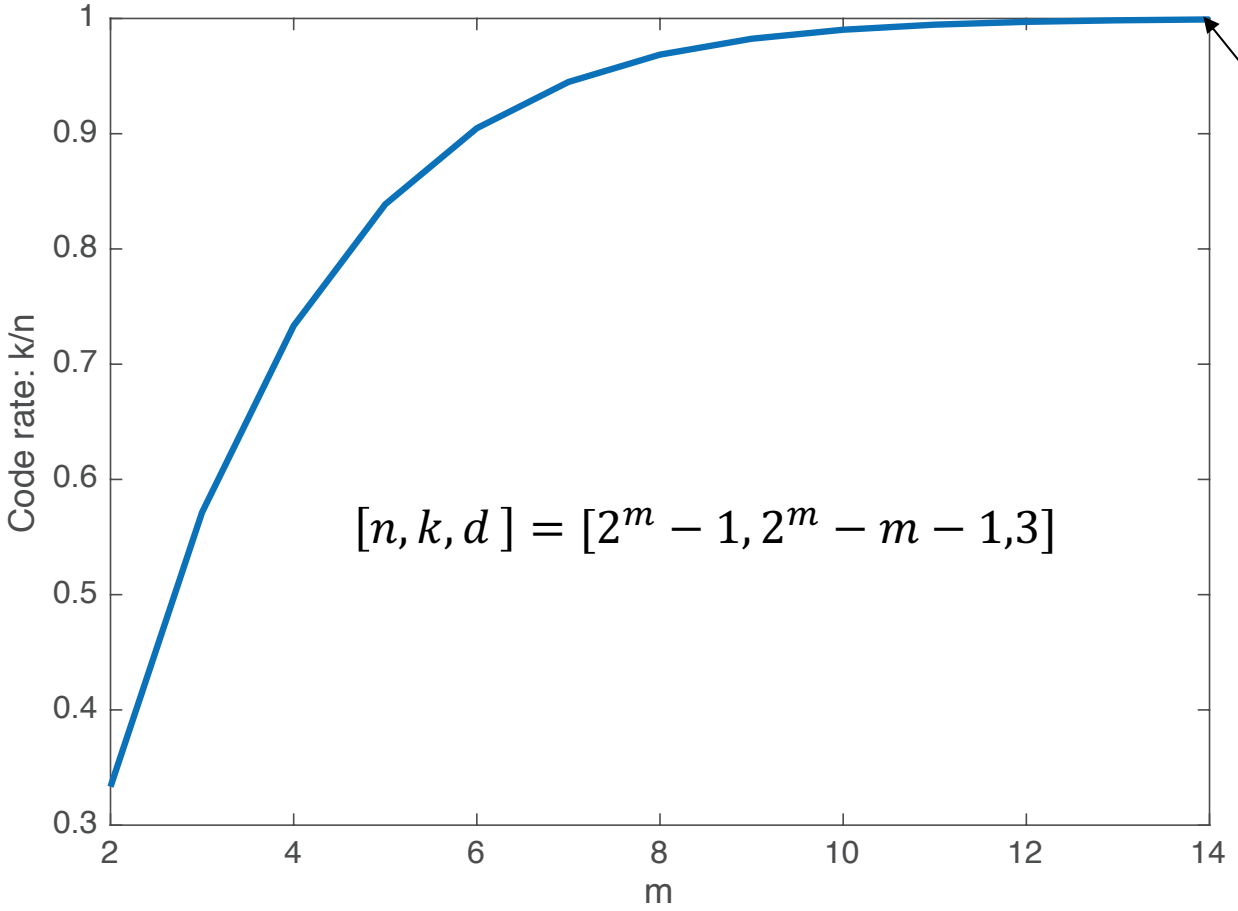
Difference Between Upper and Lower Bounds for the Maximum Value of the Minimum Distance of Binary Linear Block Codes



Data fetched from http://www.win.tue.nl/~aeb/voorlincod.html on March 14, 2005.
© 2005 Mikael Olofsson, mikael@isy.liu.se

Difference is 0 1 2 3 4 5 6 7 8

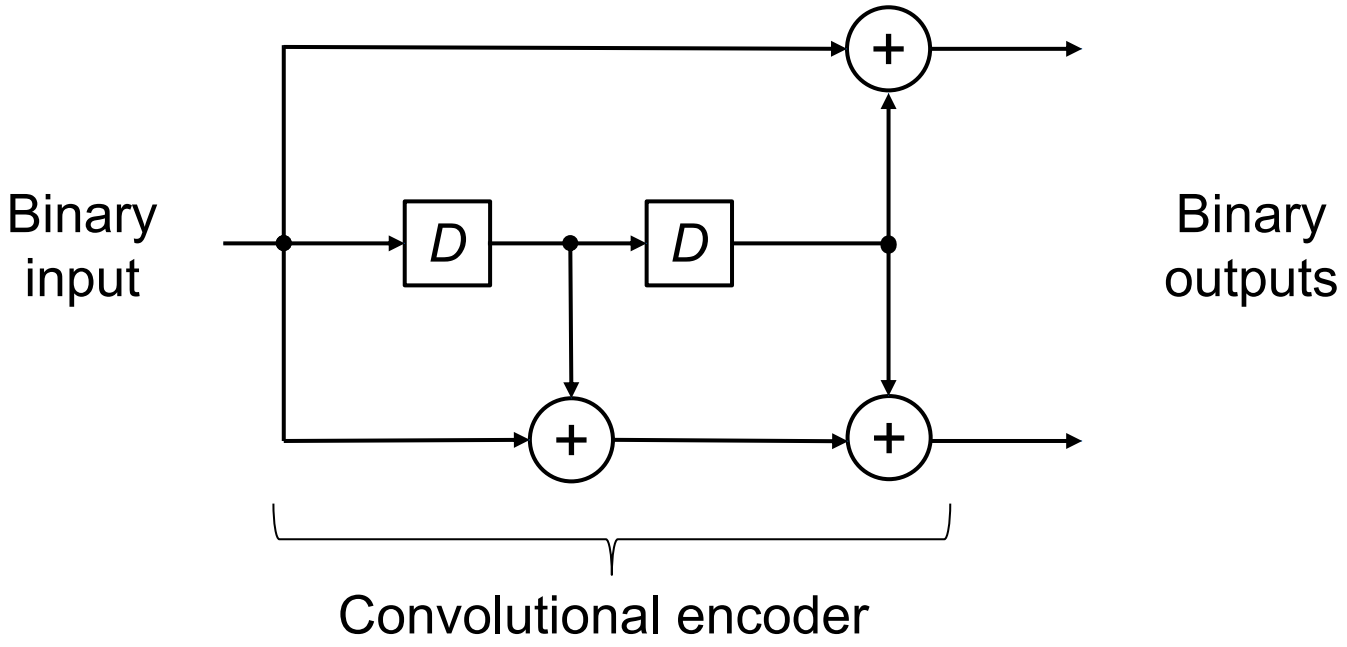
Example: Rate with Hamming code



Long block codes: Can achieve a given d with almost code rate 1

Potential drawbacks: Large delay? Complicated decoding?

Introduction: Convolutional Codes



- Encoder: FIR filter, convolution
- State diagram, trellis
- Decoding: Viterbi algorithm

Input: Semi-infinite sequence

Output: Semi-infinite sequence

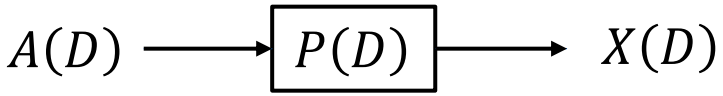
In practice: Not infinitely long

Binary Sequences

Consider: $a = a_0, a_1, a_2, \ldots$ for $a_i \in \mathbb{F}_2$

- D-transform: Write sequence using delay operator D
- $A(D) = a_0 + a_1 D + a_2 D^2 + \cdots$

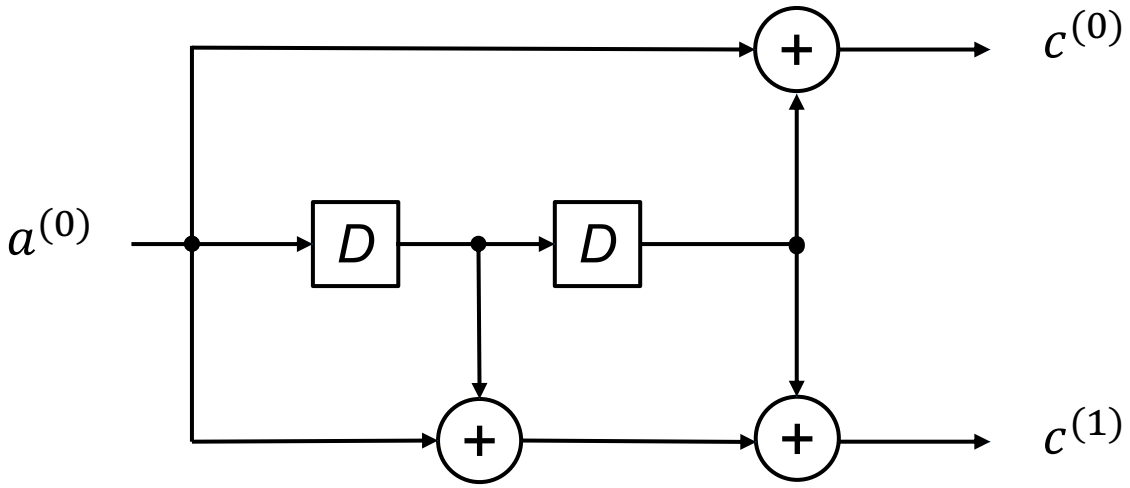
Filtering of binary sequences:



- $P(D) = p_0 + p_1 D + p_2 D^2 + \cdots + p_m D^m$
- Consider: $X(D) = A(D)P(D)$
- Time representation $x = a * p$:

$$x_m = \sum_{k=0}^{m} a_k p_{m-k} = \sum_{k=0}^{m} a_{m-k} p_k$$

Impulse Response



- Impulse response:
  - $a^{(0)} = 1,0,\ldots$ gives outputs $c^{(0)} = 1,0,1,0,\ldots$ and $c^{(1)} = 1,1,1,0,\ldots$
  - $A(D) = 1$ gives output $C^{(0)}(D) = 1 + D^2$ and $C^{(1)}(D) = 1 + D + D^2$
- Gather in a generator matrix:

$$G(D) = (1 + D^2, 1 + D + D^2)$$

Generating a Convolutional Code

- Dimensions:
  - k inputs to encoder
  - n outputs from encoder

  } Coding rate: $R = \frac{k}{n}$

- Generator matrix $G(D)$
  - Dimension $k \; x \; n$

- Generating codewords
  - Input: $A(D) = a_0 + a_1 D + a_2 D^2 + \cdots$
  - Output: $C(D) = A(D)G(D)$


LINKÖPING
UNIVERSITY
www.liu.se