ECE253/CSE208 Introduction to Information Theory

Lecture 12: Channel Coding

Dr. Yu Zhang

ECE Department
University of California, Santa Cruz

- Chap 7 of *Elements of Information Theory (2nd Edition) by Thomas Cover & Joy Thomas*.

Repetition Codes

Errors in communications can be detected and/or corrected by using

1. **Automatic Repeat Request (ARQ):** When a receiver circuit detects errors in a block of data, it requests that the block be re-transmitted.
2. **Forward Error Correction (FEC):** The transmitted data are encoded so that the receiver can correct as well as detect errors.

Repetition Codes: Introduce redundancy naively by repeating the codes a few times.

- For example, consider $\{1 \to 11111;\ 0 \to 00000\}$. Do a majority vote for decoding so that errors can be corrected as long as no more than 3 bits are corrupted. But this is not an efficient scheme since the rate is only $\frac{1}{5}$ bit per symbol.
- Repetition code cannot guarantee error-free communication, unless the effective rate goes to zero asymptotically.

Channel Coding

- The channel coding theorem promises the existence of good codes for transmitting information reliably at rates below capacity.

- The direct part implies that when the block length n is long enough, if the codewords are chosen randomly, most likely the code is good.

- A randomly constructed code has the following issues:
  1. Encoding and decoding are computationally prohibitive (exponential in n).
  2. High storage requirements for encoder and decoder.
  3. For example, if $R = \frac{1}{2}$ and $n = 1000$, we have $M = 2^{nR} = 2^{500} \approx 10^{150}$ codewords!

Channel Coding (cont'd)

- Construction of codes with efficient encoding/decoding algorithms is in the field of *channel coding theory* (study finite fields first).

- To goal is to find a capacity-achieving code with efficient encoding/decoding in terms of computation and storage.

- All channel codes used in practice are linear. Linear codes are algebraic codes, typically over a finite field, where the (symbol-wise) sum of two codewords is always a codeword and the (symbol-wise) multiplication of a codeword by a field element is also a codeword.

Type of Channel Codes

Linear codes include block codes and convolutional codes.

- **Block codes:** Map a block of info bits into a codeword, no dependence on past bits.
  A binary block code with minimum distance d can correct $\lfloor \frac{d-1}{2} \rfloor$ bit errors.
  Examples:
  1. `Hamming codes, cyclic codes, Reed-Solomon (RS) codes, Bose-Chaudhuri-Hocquenghem (BCH) codes, etc.`
     Applications: Satellite communications, storage devices such as DVD, SSD, USB, as well as high-speed applications like DVB, ADSL, xDSL.
  2. `Low-density parity-check (LDPC) codes and Polar codes.`
     Applications: 5G, DVB-S2, WiMAX, WiFi 4.

- **Convolutional codes:** Each output block depends also on some of the past inputs:
  Decoding can be efficiently implemented via Viterbi algorithm.
  Examples: `Turbo codes (used in 3G/4G, deep space communications), Trellis codes.`

Development of Channel Codes

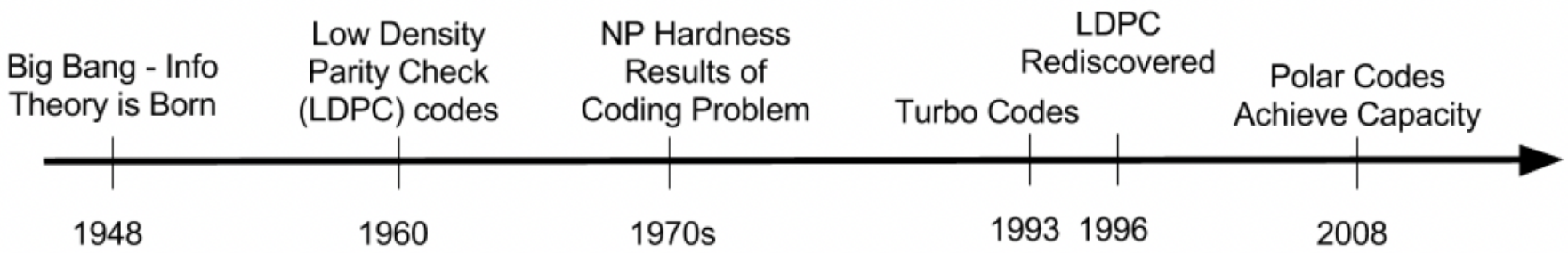


Figure: Timeline of developments towards efficient capacity-achieving codes. LDPC can reach within 0.0045 dB of the Shannon limit for binary AWGN channels. Figure credit to Prof. David Tse's lecture notes.

- The patent application for turbo codes was filed in Apr. 1991, which listed Claude Berrou as the sole inventor. The first public paper was "Near Shannon Limit Error-correcting Coding and Decoding: Turbo-code" in 1993 ICC with 3 authors: Berrou, Glavieux, and Thitimajshima.
- LDPC codes were first proposed by Robert Gallager in his 1962 PhD thesis at MIT.
- LDPC codes may be more efficient on relatively large code rates (e.g. 3/4, 5/6, 7/8) while turbo codes are the best solution at the lower code rates (e.g. 1/6, 1/3, 1/2).

Hamming Codes[1]

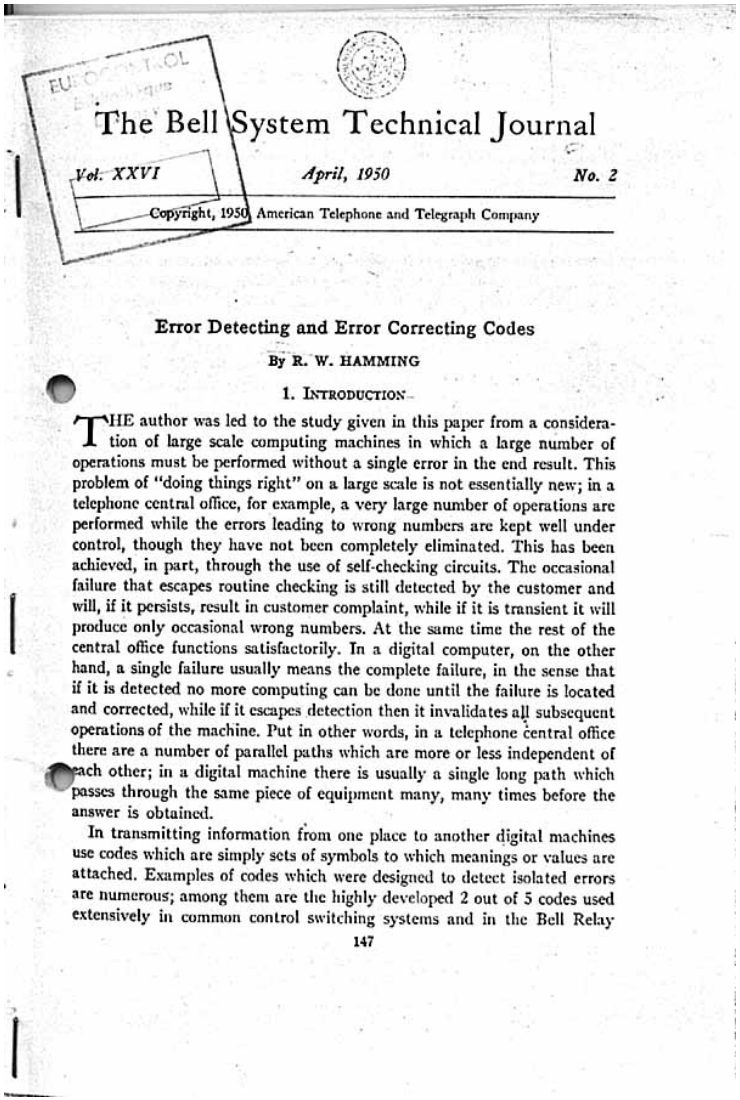
To illustrate basic ideas underlying many codes, we will focus on the Hamming codes.

Background:

- In the late 1940s, Richard Hamming worked on the Bell Model V computer, an electro-mechanical relay-based machine with cycle times in seconds. Input was fed in on punched paper tape.
- Hamming grew increasingly frustrated with having to restart his programs from scratch due to detected errors. He said in an interview, "Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?"
- Over the next few years, he worked on the problem of error-correction, developing an increasingly powerful array of algorithms. In 1950, he published what is now known as Hamming code, which remains in use today in applications such as ECC memory.

[1] `https://en.wikipedia.org/wiki/Hamming_code`

Hamming Codes (cont'd)



The Bell System Technical Journal

Vol. XXVI April, 1950 No. 2

Error Detecting and Error Correcting Codes

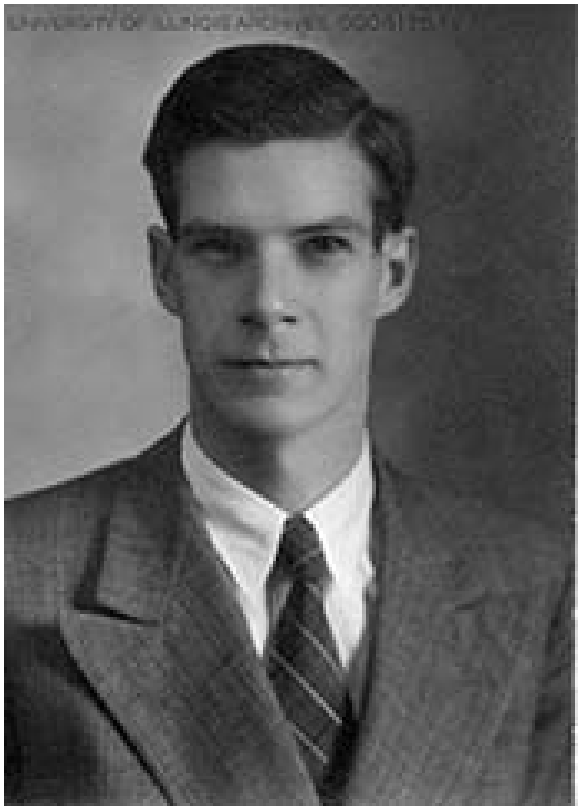
By R. W. HAMMING

1. Introduction

THE author was led to the study given in this paper from a consideration of large scale computing machines in which a large number of operations must be performed without a single error in the end result. This problem of "doing things right" on a large scale is not essentially new; in a telephone central office, for example, a very large number of operations are performed while the errors leading to wrong numbers are kept well under control, though they have not been completely eliminated. This has been achieved, in part, through the use of self-checking circuits. The occasional failure that escapes routine checking is still detected by the customer and will, if it persists, result in customer complaint, while if it is transient it will produce only occasional wrong numbers. At the same time the rest of the central office functions satisfactorily. In a digital computer, on the other hand, a single failure usually means the complete failure, in the sense that if it is detected no more computing can be done until the failure is located and corrected, while if it escapes detection then it invalidates all subsequent operations of the machine. Put in other words, in a telephone central office there are a number of parallel paths which are more or less independent of each other; in a digital machine there is usually a single long path which passes through the same piece of equipment many, many times before the answer is obtained.

In transmitting information from one place to another digital machines use codes which are simply sets of symbols to which meanings or values are attached. Examples of codes which were designed to detect isolated errors are numerous; among them are the highly developed 2 out of 5 codes used extensively in common control switching systems and in the Bell Relay

147



(7,4,3) Hamming Code

$$\mathbf{H}_{3\times 7} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \text{rank}(\mathbf{H}) = 3, \quad \dim(\text{null}(\mathbf{H})) = 4.$$

Columns of the parity check matrix $\mathbf{H}$ collect all non-zero binary vectors of length 3.

All $2^4 = 16$ null space vectors of $\mathbf{H}$ form the set of codewords that are given as follows; i.e., (all operations are done modulo-2).

$$\left\{\mathbf{c} \in \mathbb{B}^7 \mid \mathbf{Hc} = \mathbf{0}\right\}$$

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

Figure: (7,4,3) Hamming code: It encodes four data bits into seven bits by adding three parity bits. The first 4 bits of the 16 codes cycle through all 2^4 combinations of 4 bits, which can be used to represent 16 messages we want to send. The remaining 3 bits are parity check bits that are determined by the codes.

Generator Matrix

Alternatively, we can use the following generator matrix to generate an encoded sequence from the information bits.

The generator matrix $\mathbf{G}$ must satisfy $\mathbf{H}\mathbf{G}^\top = \mathbf{0}$.

$$\mathbf{G} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}_{4,7}$$

$$\vec{x} = \vec{a}G = (1 \quad 0 \quad 1 \quad 1) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = (1 \quad 0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 2) = (1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0)$$

Figure: The vector 1011 is the data to be encoded, i.e., the information we want to send.

Property of Hamming Codes

- Hamming codes are binary linear codes: sum of any two codes is also a code.
- Minimum weight is 3: minimum number of 1's in any code is 3 (except for the all-zero code).
- Minimum distance is 3: minimum number of positions in which two codes differ.

Error correction. If a code is corrupted only in one bit, it can be detected and corrected:

$$\mathbf{H}(\mathbf{c}+\mathbf{e}_i) = \mathbf{H}\mathbf{e}_i = \mathbf{H}(:,i) \ \leftarrow \ \text{the } i\text{-th column of } \mathbf{H},$$

where $\mathbf{e}_i$ is the i-th canonical vector.

It can detect and correct single-bit errors. With the addition of an overall parity bit, it can also detect (but not correct) double-bit errors.

Venn Diagram

Bit Position		1	2	3	4	5	6	7
Encoded Bits		p1	p2	d1	p4	d2	d3	d4
Parity Bit Coverage	p1	X		X		X		X
	p2		X	X			X	X
	p4				X	X	X	X

Figure: All bit positions that are powers of two are parity bits: 1, 2, 4, and etc. Each parity bit "covers" all bits where the bitwise AND of the parity and the bit position is non-zero (X marked).

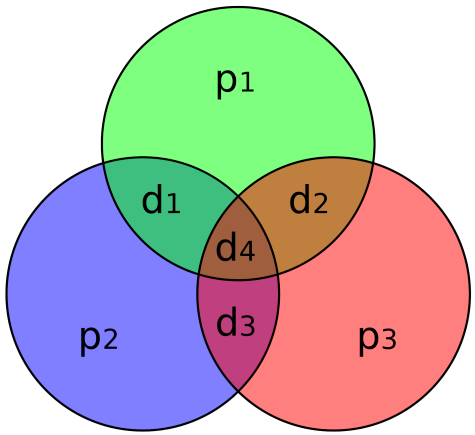


Figure: Venn diagram of the (7,4,3) Hamming code: 4 information bits (d_1, d_2, d_3, d_4) are in the intersection regions. The 3 parity bits (p_1, p_2, p_3) are added so that the parity (number of 1's) of each circle is even. If 1 info bit is flipped, the parity of at least two circles will change.

Generalization

For each integer $m \geq 2$ number of parity bits, the (n, k, d) Hamming code has

- the block length $n = 2^m - 1$
- message length $k = 2^m - m - 1$,
- minimum distance is always $d = 3$.
- Hamming code rate is $R = \frac{k}{n} = 1 - \frac{m}{2^m - 1}$

The parity check matrix $\mathbf{H}_{m \times n}$ is constructed by listing all non-zero vectors of length m.

Parity bits	Total bits	Data bits	Name	Rate
2	3	1	Hamming(3,1) (Triple repetition code)	1/3 ≈ 0.333
3	7	4	Hamming(7,4)	4/7 ≈ 0.571
4	15	11	Hamming(15,11)	11/15 ≈ 0.733
5	31	26	Hamming(31,26)	26/31 ≈ 0.839
6	63	57	Hamming(63,57)	57/63 ≈ 0.905
7	127	120	Hamming(127,120)	120/127 ≈ 0.945
8	255	247	Hamming(255,247)	247/255 ≈ 0.969
...				
m	$n = 2^m - 1$	$k = 2^m - m - 1$	Hamming$(2^m - 1, 2^m - m - 1)$	$(2^m - m - 1)/(2^m - 1)$

Figure: All the possible Hamming codes.

Thank You!

Email: <zhangy@ucsc.edu>

Homepage: https://people.ucsc.edu/~yzhan419/