Part I

Basics of coding theory

CHAPTER 1: Basics of coding theory

ABSTRACT

Coding theory - theory of error correcting codes - is one of the most interesting and applied part of mathematics and informatics.

All real communication systems that work with digitally represented data, as CD players, TV, fax machines, internet, satellites, mobiles, require to use error correcting codes because all real channels are, to some extent, noisy – due to interference caused by environment

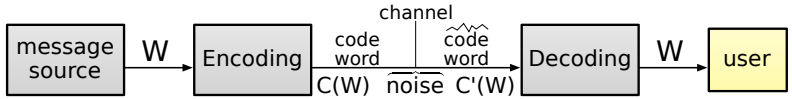
- Coding theory problems are therefore among the very basic and most frequent problems of storage and transmission of information.
- Coding theory results allow to create reliable systems out of unreliable systems to store and/or to transmit information.
- Coding theory methods are often elegant applications of very basic concepts and methods of (abstract) algebra.

This first chapter presents and illustrates the very basic problems, concepts, methods and results of coding theory.

Coding - basic concepts

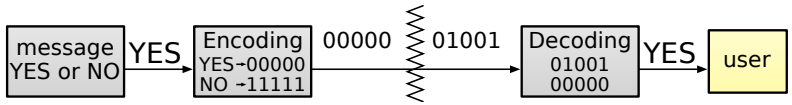
Without coding theory and error-correcting codes there would be no deep-space travel and pictures, no satellite TV, no compact disc, no ... no ... no ....

Error-correcting codes are used to correct messages when they are transmitted through noisy channels.



Error correcting framework

Example



A code C over an alphabet Σ is a subset of Σ^* ($C \subseteq \Sigma^*$).
A q-nary code is a code over an alphabet of q-symbols.
A binary code is a code over the alphabet $\{0, 1\}$.
Examples of codes $C1 = \{00, 01, 10, 11\}$ $C2 = \{000, 010, 101, 100\}$
$C3 = \{00000, 01101, 10111, 11011\}$

CHANNEL

is any physical medium through which information is transmitted.
(Telephone lines and the atmosphere are examples of channels.)

NOISE

may be caused by sunspots, lighting, meteor showers, random radio disturbance, poor typing, poor hearing, ...

TRANSMISSION GOALS

1. Fast encoding of information.
2. Easy transmission of encoded messages.
3. Fast decoding of received messages.
4. Reliable correction of errors introduced in the channel.
5. Maximum transfer of information per unit time.

BASIC METHOD OF FIGHTING ERRORS: REDUNDANCY!!!

0 is encoded as 00000 and 1 is encoded as 11111.

IMPORTANCE of ERROR-CORRECTING CODES

In a good cryptosystem a change of a single bit of the cryptotext should change so many bits of the plaintext obtained from the cryptotext that the plaintext gets uncomprehensible.

Methods to detect and correct errors when cryptotexts are transmitted are therefore much needed.

Also many non-cryptographic applications require error-correcting codes. For example, mobiles, CD-players,...

BASIC IDEA

The details of techniques used to protect information against noise in practice are sometimes rather complicated, but basic principles are easily understood.

The key idea is that in order to protect a message against a noise, we should encode the message by adding some redundant information to the message.

In such a case, even if the message is corrupted by a noise, there will be enough redundancy in the encoded message to recover – to decode the message completely.

EXAMPLE

In case of the encoding

$$0 \to 000 \quad 1 \to 111$$

the probability of the bit error $p \leq \frac{1}{2}$, and the majority voting decoding

$$000, 001, 010, 100 \to 000 \quad \text{and} \quad 111, 110, 101, 011 \to 111$$

the probability of an erroneous decoding (if there are 2 or 3 errors) is

$$3p^2(1-p) + p^3 = 3p^2 - 2p^3 < p$$

EXAMPLE: Coding of a path avoiding an enemy territory

Story Alice and Bob share an identical map (Fig. 1) gridded as shown in Fig.1. Only Alice knows the route through which Bob can reach her avoiding the enemy territory. Alice wants to send Bob the following information about the safe route he should take.

NNWNNWWSSWWNNNNWWN

Three ways to encode the safe route from Bob to Alice are:

1. $C1 = \{N = 00, W = 01, S = 11, E = 10\}$

Any error in the code word

0000010000010111110101000000000010100

would be a disaster.

2. $C2 = \{000, 011, 101, 110\}$

A single error in encoding each of symbols N, W, S, E can be detected.

3. $C3 = \{00000, 01101, 10110, 11011\}$

A single error in decoding each of symbols N, W, S, E can be corrected.

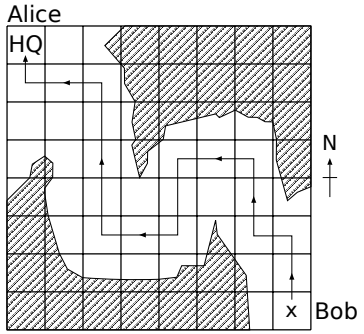


Fig. 1

Basic terminology

Block code - a code with all words of the same length.
Codewords - words of some code.

Basic assumptions about channels

1. Code length preservation Each output word of a channel has the same length as the input codeword.
2. Independence of errors The probability of any one symbol being affected in transmissions is the same.

Basic strategy for decoding

For decoding we use the so-called maximal likehood principle, or nearest neighbor decoding strategy, or majority voting decoding strategy which says that the receiver should decode a word w' as that codeword w that is the closest one to w'.

Hamming distance

The intuitive concept of "**closeness**" of two words is well formalized through Hamming distance $h(x, y)$ of words x, y. For two words x, y

h(x, y) = the number of symbols in which the words x and y differ.

Example: $h(10101, 01100) = 3$, $h(fourth, eighth) = 4$

Properties of Hamming distance

1. $h(x, y) = 0 \Leftrightarrow x = y$
2. $h(x, y) = h(y, x)$
3. $h(x, z) \leq h(x, y) + h(y, z)$ triangle inequality

An important parameter of codes C is their minimal distance.

$$h(C) = min\{h(x, y) \,|\, x, y \in C, x \neq y\},$$

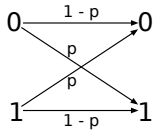
because $h(C)$ is the smallest number of errors needed to change one codeword into another.

Theorem Basic error correcting theorem
1. A code C can detect up to s errors if $h(C) \geq s + 1$.
2. A code C can correct up to t errors if $h(C) \geq 2t + 1$.

Proof (1) Trivial. (2) Suppose $h(C) \geq 2t + 1$. Let a codeword x is transmitted and a word y is recceived with $h(x, y) \leq t$. If $x' \neq x$ is a codeword, then $h(y, x') \geq t + 1$ because otherwise $h(y, x') < t + 1$ and therefore $h(x, x') \leq h(x, y) + h(y, x') < 2t + 1$ what contradicts the assumption $h(C) \geq 2t + 1$.

Binary symmetric channel

Consider a transition of binary symbols such that each symbol has probability of error $p < \frac{1}{2}$.



Binary symmetric channel

If n symbols are transmitted, then the probability of t errors is

$$p^t(1-p)^{n-t}\binom{n}{t}$$

In the case of binary symmetric channels, the "nearest neighbour decoding strategy" is also "maximum likelihood decoding strategy".
Example Consider $C = \{000, 111\}$ and the nearest neighbour decoding strategy.
Probability that the received word is decoded correctly

$$\text{as 000 is } (1-p)^3 + 3p(1-p)^2,$$
$$\text{as 111 is } (1-p)^3 + 3p(1-p)^2,$$

Therefore $P_{err}(C) = 1 - ((1-p)^3 + 3p(1-p)^2)$
is probability of erroneous decoding.
Example If $p = 0.01$, then $P_{err}(C) = 0.000298$ and only one word in 3356 will reach the user with an error.

POWER of PARITY BITS

Example Let all 2^{11} of binary words of length 11 be codewords.
Let the probability p of a bit error be 10^{-8}.
Let bits be transmitted at the rate 10^7 bits per second.
The probability that a word is transmitted incorrectly is approximately

$$11p(1-p)^{10} \approx \frac{11}{10^8}.$$

Therefore $\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1$ of words per second are transmitted incorrectly.
One wrong word is transmitted every 10 seconds, 360 erroneous words every hour and 8640 words every day without being detected!
Let now one parity bit be added.
Any single error can be detected!!!
The probability of at least two errors is:

$$1 - (1-p)^{12} - 12(1-p)^{11}p \approx \binom{12}{2}(1-p)^{10}p^2 \approx \frac{66}{10^{16}}$$

Therefore approximately $\frac{66}{10^{16}} \cdot \frac{10^7}{12} \approx 5.5 \cdot 10^{-9}$ words per second are transmitted with an undetectable error.
Corollary One undetected error occurs only every 2000 days! $(2000 \approx \frac{10^9}{5.5 \times 86400})$.

TWO-DIMENSIONAL PARITY CODE

The two-dimensional parity code arranges the data into a two-dimensional array and then to each row (column) parity bit is attached.
Example Binary string

10001011000100101111

is represented and encoded as follows

$$\begin{matrix}1&0&0&0&1\\0&1&1&0&0\\0&1&0&0&1\\0&1&1&1&1\end{matrix} \rightarrow \begin{matrix}1&0&0&0&1&0\\0&1&1&0&0&0\\0&1&0&0&1&0\\0&1&1&1&1&0\\1&1&0&1&1&0\end{matrix}$$

Question How much better is two-dimensional encoding than one-dimensional encoding?

Notation and Examples

Notation: An (n, M, d)-code C is a code such that

- n - is the length of codewords.
- M - is the number of codewords.
- d - is the minimum distance in C.

Example:
$C1 = \{00, 01, 10, 11\}$ is a (2,4,1)-code.
$C2 = \{000, 011, 101, 110\}$ is a (3,4,2)-code.
$C3 = \{00000, 01101, 10110, 11011\}$ is a (5,4,3)-code.

Comment: A good (n, M, d)-code has small n and large M and d.

Examples from deep space travels

Examples (Transmission of photographs from the deep space)

- In 1965-69 **Mariner 4-5** took the first photographs of another planet - 22 photos. Each photo was divided into 200 × 200 elementary squares - pixels. Each pixel was assigned 6 bits representing 64 levels of brightness. Hadamard code was used.

  Transmission rate: 8.3 bits per second.
- In 1970-72 **Mariners 6-8** took such photographs that each picture was broken into 700 × 832 squares. Reed-Muller (32,64,16) code was used.

  Transmission rate was 16200 bits per second. (Much better pictures)

HADAMARD CODE

In Mariner 5, 6-bit pixels were encoded using 32-bit long Hadamard code that could correct up to 7 errors.

Hadamard code has 64 codewords. 32 of them are represented by the 32 × 32 matrix $H = \{h_{IJ}\}$, where $0 \leq i, j \leq 31$ and

$$h_{ij} = (-1)^{a_0 b_0 + a_1 b_1 + \ldots + a_4 b_4}$$

where i and j have binary representations

$$i = a_4 a_3 a_2 a_1 a_0,\ j = b_4 b_3 b_2 b_1 b_0$$

The remaing 32 codewords are represented by the matrix $-H$.
Decoding is quite simple.

CODE RATE

For q-nary (n, M, d)-code we define code rate, or information rate, R, by

$$R = \frac{lg_q M}{n}.$$

The code rate represents the ratio of the number of needed input data symbols to the number of transmitted code symbols.

Code rate (6/32 for Hadamard code), is an important parameter for real implementations, because it shows what fraction of the bandwidth is being used to transmit actual data.

The ISBN-code

Each book till 1.1.2007 had International Standard Book Number which was a 10-digit codeword produced by the publisher with the following structure:

l	p	m	w	$= x_1 \dots x_{10}$
language	publisher	number	weighted check sum	
0	07	709503	0	

such that $\sum_{i=1}^{10} ix_i \equiv 0 \ (mod 11)$

The publisher had to put X into the 10-th position if $x_{10} = 10$.
The ISBN code was designed to detect: (a) any single error (b) any double error created by a transposition

Single error detection

Let $X = x_1 \dots x_{10}$ be a correct code and let

$$Y = x_1 \dots x_{J-1} y_J x_{J+1} \dots x_{10} \text{ with } y_J = x_J + a, a \neq 0$$

In such a case:

$$\sum_{i=1}^{10} iy_i = \sum_{i=1}^{10} ix_i + ja \neq 0 \ (mod 11)$$

The ISBN-code

Transposition detection

Let x_J and x_k be exchanged.

$$\sum_{i=1}^{10} iy_i = \sum_{i=1}^{10} ix_i + (k-j)x_j + (j-k)x_k = (k-j)(x_j - x_k) \neq 0 \ (mod 11)$$

if $k \neq j$ and $x_j \neq x_k$.

New ISBN code

Starting 1.1.2007 instead of 10-digit ISBN code a 13-digit ISBN code is being used.

New ISBN number can be obtained from the old one by preceeding the old code with three digits 978.

For details about 13-digit ISBN see

`http://www.isbn-international.org/en/revision.html`

Equivalence of codes

Definition Two q-ary codes are called equivalent if one can be obtained from the other by a combination of operations of the following type:

(a) a permutation of the positions of the code.

(b) a permutation of symbols appearing in a fixed position.

Question: Let a code be displayed as an M $\times$ n matrix. To what correspond operations (a) and (b)?
Claim: Distances between codewords are unchanged by operations (a), (b).
Consequently, equivalent codes have the same parameters (n,M,d) (and correct the same number of errors).

Examples of equivalent codes

$$(1)\ \begin{Bmatrix} 0&0&1&0&0\\ 0&0&0&1&1\\ 1&1&1&1&1\\ 1&1&0&0&0 \end{Bmatrix} \begin{Bmatrix} 0&0&0&0&0\\ 0&1&1&0&1\\ 1&0&1&1&0\\ 1&1&0&1&1 \end{Bmatrix} \quad (2)\ \begin{Bmatrix} 0&0&0\\ 1&1&1\\ 2&2&2 \end{Bmatrix} \begin{Bmatrix} 0&1&2\\ 1&2&0\\ 2&0&1 \end{Bmatrix}$$

Lemma Any q-ary (n, M, d)-code over an alphabet $\{0, 1, \ldots, q-1\}$ is equivalent to an (n, M, d)-code which contains the all-zero codeword $00\ldots0$.
Proof Trivial.

The main coding theory problem

A good (n, M, d)-code has small n, large M and large d.

The main coding theory problem is to optimize one of the parameters n, M, d for given values of the other two.

Notation: $A_q(n, d)$ is the largest M such that there is an q-nary (n, M, d)-code.

Theorem
(a) $A_q(n, 1) = q^n$;
(b) $A_q(n, n) = q$.

Proof

(a) obvious;

(b) Let C be an q-nary (n, M, n)-code. Any two distinct codewords of C differ in all n positions. Hence symbols in any fixed position of M codewords have to be different $\Rightarrow A_q(n, n) \leq q$. Since the q-nary repetition code is (n, q, n)-code, we get $A_q(n, n) \geq q$.

EXAMPLE

Example Proof that $A_2(5, 3) = 4$.

(a) Code C_3 is a $(5, 4, 3)$-code, hence $A_2(5, 3) \geq 4$.

(b) Let C be a $(5, M, 3)$-code with $M = 5$.

- By previous lemma we can assume that $00000 \in C$.
- C has to contain at most one codeword with at least four 1's. (otherwise $d(x, y) \leq 2$ for two such codewords x, y)
- Since $00000 \in C$, there can be no codeword in C with at most one or two 1.
- Since $d = 3$, C cannot contain three codewords with three 1's.
- Since $M \geq 4$, there have to be in C two codewords with three 1's. (say 11100, 00111), the only possible codeword with four or five 1's is then 11011.

Design of one code from another code

Theorem Suppose d is odd. Then a binary (n, M, d)-code exists if a binary $(n+1, M, d+1)$-code exists.

Proof Only if case: Let C be a binary (n, M, d) code. Let

$$C' = \{x_1 \ldots x_n x_{n+1} | x_1 \ldots x_n \in C, x_{n+1} = (\textstyle\sum_{i=1}^{n} x_i)\ mod\ 2\}$$

Since parity of all codewords in C' is even, $d(x', y')$ is even for all

$$x', y' \in C'.$$

Hence $d(C')$ is even. Since $d \leq d(C') \leq d+1$ and d is odd,

$$d(C') = d+1.$$

Hence C' is an $(n+1, M, d+1)$-code.

If case: Let D be an $(n+1, M, d+1)$-code. Choose code words x, y of D such that $d(x, y) = d+1$.
Find a position in which x, y differ and delete this position from all codewords of D.
Resulting code is an (n, M, d)-code.

A corollary

Corollary:
If d is odd, then $A_2(n,d) = A_2(n+1,d+1)$.
If d is even, then $A_2(n,d) = A_2(n-1,d-1)$.

Example $A_2(5,3) = 4 \Rightarrow A_2(6,4) = 4$
$(5,4,3)$-code $\Rightarrow$ $(6,4,4)$-code

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{array}$$ by adding check.

A sphere and its contents

Notation F_q^n - is a set of all words of length n over the alphabet $\{0,1,2,\ldots,q-1\}$

Definition For any codeword $u \in F_q^n$ and any integer $r \geq 0$ the sphere of radius r and centre u is denoted by

$$S(u,r) = \{v \in F_q^n | d(u,v) \leq r\}.$$

Theorem A sphere of radius r in F_q^n, $0 \leq r \leq n$ contains

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \ldots + \binom{n}{r}(q-1)^r$$

words.

Proof Let u be a fixed word in F_q^n. The number of words that differ from u in m position is

$$\binom{n}{m}(q-1)^m.$$

General upper bounds

Theorem **(The sphere-packing or Hamming bound)**
If C is a q-nary $(n, M, 2t+1)$-code, then

$$M\left\{\binom{n}{0} + \binom{n}{1}(q-1) + \ldots + \binom{n}{t}(q-1)^t\right\} \leq q^n \quad (1)$$

Proof Any two spheres of radius t centred on distinct codewords have no codeword in common. Hence the total number of words in M spheres of radius t centred on M codewords is given by the left side (1). This number has to be less or equal to q^n.

A code which achieves the sphere-packing bound from (1), i.e. such a code that equality holds in (1), is called a **perfect code**.

Singleton bound: If C is an q-ary (n, M, d) code, then

$$M \leq q^{n-d+1}$$

A general upper bound on $A_q(n,d)$

Example An $(7, M, 3)$-code is perfect if

$$M\left(\binom{7}{0} + \binom{7}{1}\right) = 2^7$$

i.e. $M = 16$

An example of such a code:

$C4 = \{0000000, 1111111, 1000101, 1100010, 0110001, 1011000, 0101100,$
$0010110, 0001011, 0111010, 0011101, 1001110, 0100111, 1010011, 1101001, 1110100\}$

Table of $A_2(n,d)$ from 1981

n	$d=3$	$d=5$	$d=7$
5	4	2	-
6	8	2	-
7	16	2	2
8	20	4	2
9	40	6	2
10	72-79	12	2
11	144-158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2560-3276	256-340	36-37

For current best results see `http://www.win.tue.nl/math/dw/voorlincod.html`

LOWER BOUND for $A_q(n, d)$

The following lower bound for $A_q(n, d)$ is known as Gilbert-Varshanov bound:

Theorem Given $d \leq n$, there exists a q-ary (n, M, d)-code with

$$M \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j}$$

and therefore

$$A_q(n, d) \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j}$$

Error Detection

Error detection is much more modest aim than error correction.

Error detection is suitable in the cases that channel is so good that probability of error is small and if an error is detected, the receiver can ask to renew the transmission.

For example, two main requirements for many telegraphy codes used to be:

- Any two codewords had to have distance at least 2;
- No codeword could be obtained from another codeword

by transposition of two adjacent letters.

Pictures of Saturn taken by Voyager

Pictures of Saturn taken by Voyager, in 1980, had 800 $\times$ 800 pixels with 8 levels of brightness.

Since pictures were in color, each picture was transmitted three times; each time through different color filter. The full color picture was represented by

$$3 \times 800 \times 800 \times 8 = 13360000 \text{ bits.}$$

To transmit pictures Voyager used the Golay code G_{24}.

General coding problem

Important problems of information theory are how to define formally such concepts as information and how to store or transmit information efficiently.

Let X be a random variable (source) which takes any value x with probability $p(x)$. The entropy of X is defined by

$$S(X) = -\sum_x p(x) lg\ p(x)$$

and it is considered to be the information content of X.

In a special case of a binary variable X which takes on the value 1 with probability p and the value 0 with probability $1 - p$

$$S(X) = H(p) = -p\ lg\ p - (1-p) lg(1-p)$$

Problem: What is the minimal number of bits needed to transmit n values of X?
Basic idea: To encode more probable outputs of X by shorter binary words.
Example (Morse code - 1838)

a .-	b -...	c -.-.	d -..	e .	f ..-.	g –.
h	i ..	j .—	k -.-	l .-..	m –	n -.
o —	p .–.	q –.-	r .-.	s ...	t -	u ..-
v ...-	w .–	x -..-	y -.–	z –..		

Shannon's noisless coding theorem

Shannon's noiseless coding theorem says that in order to transmit n values of X, we need, and it is sufficient, to use $nS(X)$ bits.

More exactly, we cannot do better than the bound $nS(X)$ says, and we can reach the bound $nS(X)$ as close as desirable.

Example Let a source X produce the value 1 with probability $p = \frac{1}{4}$ and the value 0 with probability $1 - p = \frac{3}{4}$
Assume we want to encode blocks of the outputs of X of length 4.

By Shannon's theorem we need $4H(\frac{1}{4}) = 3.245$ bits per blocks (in average)

A simple and practical method known as **Huffman code** requires in this case 3.273 bits per a 4-bit message.

mess.	code	mess.	code	mess.	code	mess.	code
0000	10	0100	010	1000	011	1100	11101
0001	000	0101	11001	1001	11011	1101	111110
0010	001	0110	11010	1010	11100	1110	111101
0011	11000	0111	1111000	1011	111111	1111	1111001

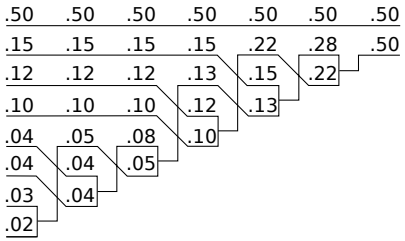
Observe that this is a **prefix code** - no codeword is a prefix of another codeword.

Design of Huffman code

Given a sequence of n objects, $x_1, \ldots, x_n$ with probabilities $p_1 \geq \ldots \geq p_n$.

Stage 1 - shrinking of the sequence.

- Replace x_{n-1}, x_n with a new object y_{n-1} with probability $p_{n-1} + p_n$ and rearrange sequence so one has again non-increasing probabilities.
- Keep doing the above step till the sequence shrinks to two objects.



Stage 2 - extending the code - Apply again and again the following method.

If $C = \{c_1, \ldots, c_r\}$ is a prefix optimal code for a source S_r, then $C' = \{c'_1, \ldots, c'_{r+1}\}$ is an optimal code for S_{r+1}, where

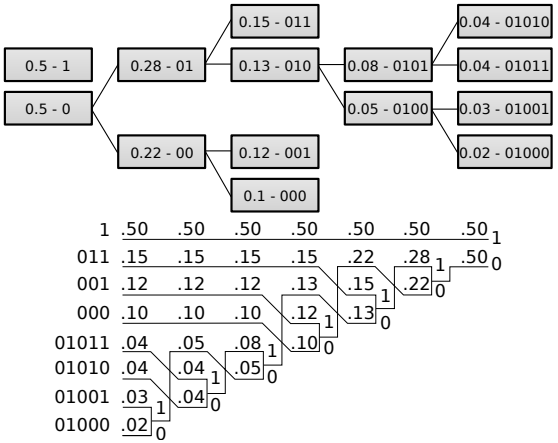
$$c'_i = c_i \quad 1 \leq i \leq r-1$$
$$c'_r = c_r 1$$
$$c'_{r+1} = c_r 0.$$

Design of Huffman code

Stage 2 Apply again and again the following method:

If $C = \{c_1, \ldots, c_r\}$ is a prefix optimal code for a source S_r, then $C' = \{c'_1, \ldots, c'_{r+1}\}$ is an optimal code for S_{r+1}, where

$$c'_i = c_i \quad 1 \leq i \leq r-1$$
$$c'_r = c_r 1$$
$$c'_{r+1} = c_r 0.$$



A BIT OF HISTORY

The subject of error-correcting codes arose originally as a response to practical problems in the reliable communication of digitally encoded information.

The discipline was initiated in the paper

Claude Shannon: A mathematical theory of communication, Bell Syst.Tech. Journal V27, 1948, 379-423, 623-656

Shannon's paper started the scientific discipline **information theory** and **error-correcting codes** are its part.

Originally, information theory was a part of electrical engineering. Nowadays, it is an important part of mathematics and also of informatics.

SHANNON's VIEW

In the introduction to his seminal paper "A mathematical theory of communication" Shannon wrote:

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.