Part I

Basics of coding theory

ABSTRACT - September 24, 2014

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

Goals of coding theory are to develop systems and methods that allow to detect/correct errors caused when information is transmitted through **noisy channels**.

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 various interference/destruction caused by the environment

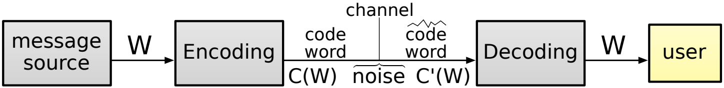
- Coding theory problems are therefore among the very basic and most frequent problems of storage and transmission of information.
- Coding theory results allow to create reliable systems out of unreliable systems to store and/or to transmit information.
- Coding theory methods are often elegant applications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

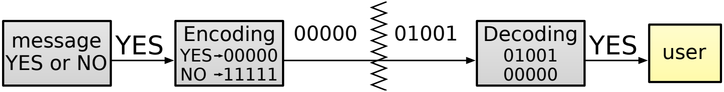
Without coding theory and error-correcting codes there would be no deep-space travel and pictures, no satellite TV, no compact disc, no ... no ... no ....

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



Error correcting framework

Example



A code C over an alphabet Σ is a subset of $\Sigma^*(C \subseteq \Sigma^*)$.
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 in which information is stored or through which information is transmitted.
(Telephone lines, optical fibres and also 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. To encode information fast.
2. Very similar messages should be encoded very differently
3. Easy transmission of encoded messages.
4. Fast decoding of received messages.
5. Reliable correction of errors introduced in the channel.
6. Maximum transfer of information per unit time.

BASIC METHOD OF FIGHTING ERRORS: REDUNDANCY!!!

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

CHANNELS - MATHEMATICAL VIEWS

Formally, a channel is described by a triple $C = (\Sigma, \Omega, p)$, where

- Σ is an input alphabet
- Ω is an output alphabet
- p is a probability distribution on $\Sigma \times \Omega$ and for $i \in \Sigma, o \in \Omega$, $p(i, o)$ **is the probability that the output of the channel is o if the input is i.**

IMPORTANT CHANNELS

- **Binary symmetric channel** maps, with fixed probability p_0, each binary input into oposite one. Hence, $Pr(0,1) = Pr(1,0) = p_0$ and $Pr(0,0) = Pr(1,1) = 1 - p_0$.
- **Binary erasure channel** maps, with fixed probability p_0, binary inputs into $\{0, 1, e\}$, where e is so called the erasure symbol, and $Pr(0,0) = Pr(1,1) = p_0$, $Pr(0, e) = Pr(1, e) = 1 - p_0$.
- **White noise Gaussian channel** that models errors in the deep space.

BASIC CHANNEL CODING PROBLEMS

Summary: The task of a channel coding is to encode the information sent over a communication channel in such a way that in the presence of some channel noise, errors can be detected and/or corrected.

There are two basic coding methods

BEC (Backward Error Correction) Coding allows the receiver only to detect errors. If an error is detected, then the sender is requested to retransmit the message.

FEC (Forward Error Correction) Coding allows the receiver to correct a certain amount of errors.

WHY WE NEED TO IMPROVE ERROR-CORRECTING CODES

When error correcting capabilities of some code are improved - that is a better code is found - this has the following impacts:

- For the same quality of the received information, it is possible to achieve that the transmission system operates in more severe conditions;
- For example;
  1. It is possible to reduce the size of antennas or solar panels and the weight of batteries;
  2. In the space travel systems such savings can be measured in hundred of thousands of dollars;
  3. In mobile telephone systems, improving the code enables the operators to increase the potential number of users in each cell.
- Another field of applications of error-correcting codes is that of mass memories: computer hard drives, CD-ROMs, DVDs and so on.

BASIC IDEA of ERROR CORRECTION

Details of the techniques used to protect information against noise in practice are sometimes rather complicated, but basic principles are mostly 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.

MAJORITY VOTING DECODING - BASIC IDEA

The basic idea of so called majority voting decoding/principle or of maximal likelihood decoding/principle is

to decode a received message w'

by a codeword w that is the closest one to w'

in the whole set of the potential codewords of a given code C.

EXAMPLE

In case: (a) the encoding

$$0 \rightarrow 000 \quad 1 \rightarrow 111,$$

is used, (b) the probability of the bit error is $p < \frac{1}{2}$, and (c) the following majority voting decoding

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

is used, then the probability of an erroneous decoding (for the case of 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. $C1 = \{N = 00, W = 01, S = 11, E = 10\}$

In such a case any error in the code word

000001000001011111010100000000010100

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.

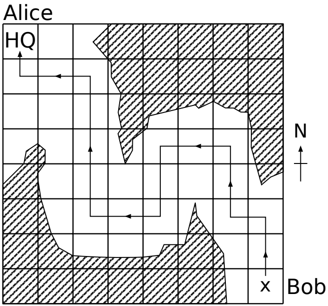


Fig. 1

BASIC TERMINOLOGY

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

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 by an error 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 received word w'

as

the 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.

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

Therefore, $h(C)$ is the smallest number of errors that can change one codeword into another.

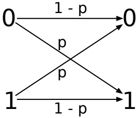
Basic error correcting theorem

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 received with $h(x, y) \leq t$. If $x' \neq x$ is any 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

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

Therefore $$P_{err}(C) = 1 - ((1-p)^3 + 3p(1-p)^2)$$
is the probability of an 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
and let the probability of a bit error be $p = 10^{-8}$.
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.
Therefore, 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 once 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

$$1000101100010010111 1$$

is represented and encoded as follows

$$\begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{array} \rightarrow \begin{array}{cccccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{array}$$

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

NOTATIONS 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, large M and also large d.

EXAMPLES from DEEP SPACE TRAVELS

Examples (Transmission of photographs from the deep space)

- In 1965-69 **Mariner 4-5** probes 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. and so called Hadamard code was used.

  Transmission rate was 8.3 bits per second.

- In 1970-72 **Mariners 6-8** took such photographs that each picture was broken into 700×832 squares. So called Reed-Muller (32,64,16) code was used.

  Transmission rate was 16200 bits per second. (Much better quality pictures could be received)

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 remaining 32 codewords are represented by the matrix $-H$.
Decoding was quite simple.

CODES RATES

For q-nary (n, M, d)-code we define the 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.

If a q-nary code has code rate R, then we say that it transmits R q-symbols per a channel use - or R is a number of bits per a channel use (bpc) - in the case of binary alphabet.

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

The ISBN-code I

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 $\sum_{i=1}^{10}(11-i)x_i \equiv 0 \ (mod 11)$

The publisher has to put $x_{10} = X$ if x_{10} is to be 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:

$$\sum_{i=1}^{10}(11-i)y_i = \sum_{i=1}^{10}(11-i)x_i + (11-j)a \neq 0 \ (mod 11)$$

The ISBN-code II

Transposition detection

Let x_j and x_k be exchanged.

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

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 preceding the old code with three digits 978.

For details about 13-digit ISBN see

`http://www.en.wikipedia.org/Wiki/International_Standard_Book_Number`

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 should have a 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) First claim is obvious;

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

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

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

$$C' = \left\{x_1 \dots x_n x_{n+1} \middle| x_1 \dots x_n \in C, x_{n+1} = \left(\sum_{i=1}^{n} x_i\right) \mathit{mod}\ 2\right\}$$

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

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

A SPHERE and its VOLUME

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

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

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 positions is

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

GENERAL UPPER BOUNDS on CODE PARAMETERS

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

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

Proof Since minimal distance of the code C is $2t+1$, 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 in (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

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

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.codetables.de

LOWER BOUND for $A_q(n,d)$

The following lower bound for $A_q(n,d)$ is known as Gilbert-Varshamov 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 an error is small and if an error is detected, the receiver can ask the sender 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, 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 so called 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 $1 - p$

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

Problem: What is the minimal number of bits needed to transmit n values of X?
Basic idea: To encode more (less) probable outputs of X by shorter (longere) 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 –..		

[1] Notation lg (ln) [log] will be used for binary, natural and decimal logarithms.

SHANNON's NOISELESS 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

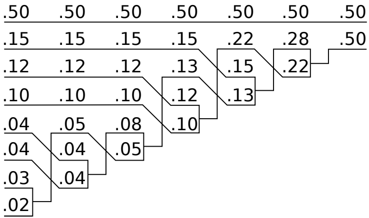
Observe that this is a **prefix code** - no codeword is a prefix of another codeword.

DESIGN of HUFFMAN CODE II

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

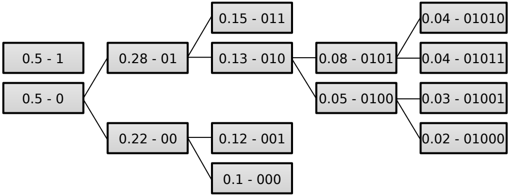
$$c'_i = c_i \quad 1 \leq i \leq r-1$$
$$c'_r = c_r 1$$
$$c'_{r+1} = c_r 0.$$

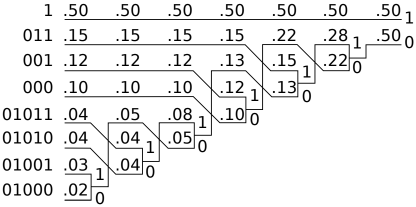
DESIGN of HUFFMAN CODE II

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 I

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

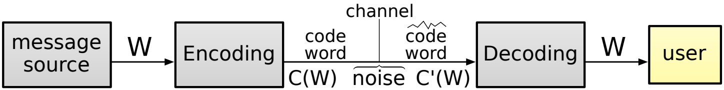
In the introduction to his seminal paper "A mathematical theory of communication" Shannon wrote:

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

APPENDIX

HARD VERSUS SOFT DECODING I

At the beginning of this chapter the process **encoding-channel transmission-decoding** was illustrated as follows:

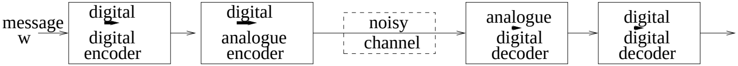


In that process a binary message is at first encoded into a binary codeword, then transmitted through a noisy channel, and, finally, the decoder receives, for decoding, a potentially erroneous binary message and makes an error correction.

This is a simplified view of the whole process. **In practice the whole process looks quite differently.**

HARD versus SOFT DECODING II

Here is a more realistic view of the whole encoding-transmission-decoding process



that is

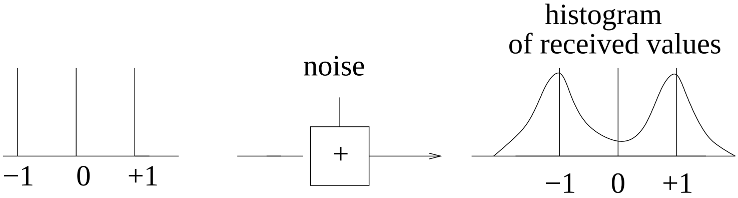
- a binary message is at first transferred to a binary codeword;
- the binary codeword is then transferred to an analogoe signal;
- the analogoe signal is then transmitted through a noisy channel
- the received analogous signal is then transferred to a binary form that can be used for decoding and, finally
- decoding takes place.

In case the analogous noisy signal is transferred before decoding to the binary signal we talk about a **hard decoding**;
In case the output of analogous-digital deoding is a pair (p_b, b) where p_b is the probability that the output is the bit b (or a weight of such a binary output (often given by a number from an interval $(-V_{max}, V_{max})$), we talk about a **soft decoding**.

HARD versus SOFT DECODING III

In order to deal with such a more general model of transmission and soft decoding, it is common to use, instead of the binary symbols 0 and 1 so-called **antipodal binary symbols** $+1$ and -1 that are represented electronically by voltage $+1$ and -1.

A transmission channel with analogue antipodal signals can then be depicted as follows.



A very important case in practise, especially for space communication, is so-called additive white Gaussian noise (AWGN) and the channel with such a noise is called Gaussian channel.

HARD versus SOFT DECODING - COMMENTS

When the signal received by the decoder comes from a devise capable of producing estimations of an analogue nature on the binary trasmitted data the error correction capability of the decoder can greatly be improved.

Since the decoder has in such a case an information about the reliability of data received, decoding on the basis of finding the codeword with minimal Hamming distance does not have to be optimal and the optimal decoding may depend on the type of noise involved.

For example, in an important practical case of the Gaussian white noise one search at the minimal likehood decoding for a codeword with minimal Euclidean distance.

BASIC FAMILIES of CODES

Two basic families of codes are

Block codes called also as **algebraic codes** that are appropriate to encode blocks of date of the same length and independent one from the other. Their encoders have often a huge number of internal states and decoding algorithms are based on techniques specific for each code.

Stream codes called also as **convolution codes** that are used to protect continuous flows of data. Their encoders often have only small number of internal states and then decoders can use a complete representation of states using so called *trellises*, iterative approaches via several simple decoders and an exchange of information of probabilistic nature.

Hard decoding is used mainly for block codes and soft one for stream codes. However, distinctions between these two families of codes are tending to blur.

NOTATIONAL COMMENT

The term code is often used also to denote a specific encoding algorithm that transfers any dataword, say of the size k, into a codeword, say of the size n. The set of all such codewords then forms the code in the original sense.

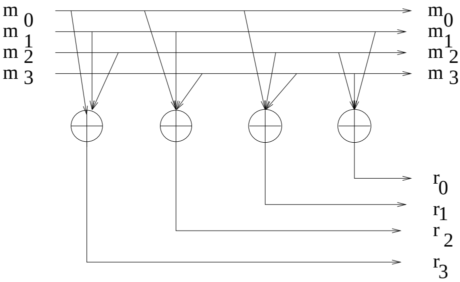
For the same code there can be many encoding algorithms that map the same set of datawords into different codewords.

SYSTEMATIC CODES I

A code is called systematic if its encoder transmit a message (an input dataword) w into a codeword of the form wc_w, or (w, c_w). That is if the codeword for the dataword w consists of two parts: dataword w (called also information part) and redundancy part c_w

Nowadays most of the stream codes that are used in practice are systematic.

An example of a systematic encoder, that produces so called extended Hamming $(8, 4, 1)$ code is in the following figure.



STORY of MORSE TELEGRAP - I.

- In 1825 William Sturgeon discovered electromagnet and showed that using electricity one can make to ring a ring that was far away.
- The first telegraph designed Charles Wheate Stone and demonstrated it at the distance 2.4 km.
- Samuel Morse made a significant improvement by designing a telegraph that could not only send information, but using a magnet at other end it could also write the transmitted symbol on a paper.
- Morse was a portrait painter whose hobby were electrical machines.
- Morse and his assistant Alfred Vailem invented "Morse alphabet" around 1842.
- After US Congress approved 30,000 $ on 3.3.1943 for building a telegraph connection between Washington and Baltimore the line was built fast and already on 24.3.1943 the first telegraph message was sent: "What hat God wrought" - "Čo Boh vykonal".
- The era of Morse telegraph ended on 26.1.2006 when Western Union announced cancelation of all telegraph services.

In his telegraphs Moorse used the following two-character audio alphabet

- TIT or dot — a short tone lasting four hundredths of second;
- TAT or dash — a long tone lasting twelve hundredth of second.

Morse could called these tones as 0 and 1

The binary elements 0 and 1 were first called bits by J. W. Tuckley in 1943.