ANALELE UNIVERSITĂȚII
"EFTIMIE MURGU" REŞIȚA
ANUL XX, NR. 1, 2013, ISSN 1453 - 7397

Silviu Drăghici, Cornelia Anghel Drugărin, Cristian Chioncel

Optimal Encoding of Data in Data Transmission Channels

This paper aims to present the methods of achieving an optimal encoding in the data communication channels. After a short description of the communication channel and of the data communication channel types, follow briefly a few notions of the data channel enthropy, information, transinformation, with their properties, definitions and mathematical relations connecting them. Chapter 2 presents the concept of optimal code, following a detailed description (using two suggestive examples) of the two main methods used to obtain an optimal code: Shannon-Fano and Huffman.

Keywords: *optimal code, communication channel, entropy, information, transition functions.*

1. Introduction

A communication channel generally presents the following block structure:

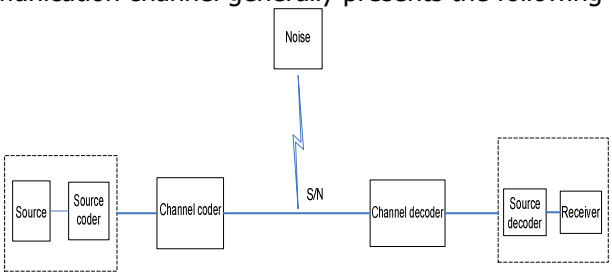

Figure 1. A communication channel

As noticed, every communication channel attaches errors, i.e. noises, characterized by the relation signal/noise (denoted S/N or SNR). The channel decoder will perform their notification and correction.

In a broader sense, a channel can be defined as any device that sends information from a transmitter (source) to a receiver. There is a variety of

communication channels. But this diversity is mathematically described by a probabilistic model that includes the following sets and variables:

1. A set of signals applied at the channel input, usually represented with **X**:

$$X = \{x_1, x_2, ..., x_m\};$$

2. A set of signals from the channel output, usually represented with **Y**:

$$Y = \{y_1, y_2, ..., y_n\};$$

3. A set of transition probabilities $p(y_j/x_i)$, represented as follows:

$\{X, Y, p(Y / X)\}$, where $p(y_j / x_i)$ represents the probability of the output variable y_j conditioned by the input variable x_i, where $i = \overline{1,m}$, and $j = \overline{1,n}$; p (Y / X) is a transition matrix of m x n size, structured as follows:

$$p(Y / X) = \begin{pmatrix} p(y_1 / x_1) & p(y_2 / x_1) & ... & p(y_n / x_1) \\ ... & ... & ... & ... \\ p(y_1 / x_m) & p(y_2 / x_m) & ... & p(y_n / x_m) \end{pmatrix} \quad (1)$$

Also, we notice that any form of energy can be used in transmitting information.
A transmission channel mandatorily contains the source and the source coder.
It is often likely to transmit correctly the information through an interference (noise) channel, depending on the selected encoding.

We notice that according to the characteristic of the sets X and Y and of the transition matrix p (Y / X), the communication channels can be classified as follows:

a) a continuous communication channel: if the set of X and Y signals are continuous;

b) discrete communication channel: if the set of X and Y signals are discrete ;

c) symmetric communication channel: if the columns as well as the lines of the transition matrix p (Y / X) can be divided in subsets so that every column of the matrix should represent a circular permutation of another column in the transition matrix; the same goes for the transition matrix lines.

The most simple and suggestive example is that of a symmetric binary channel, where the sets X, Y and the transition matrix are represented as:

$$X = \{x_1, x_2\},\ Y = \{y_1, y_2\};$$
$$p(y_1 / x_1) = p(y_2 / x_2) = 1 - p \quad (2)$$
$$p(y_2 / x_1) = p(y_1 / x_2) = p,$$

where p represents the probability for a symbol to be transmitted erroneously; p is a probability that depends on the physical parameters of the data transmission channel.

The transition matrix will have the following representation:

$$p(Y/X) = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \quad (3)$$

obviously, at the *symmetric* channel, the sum of the transition probabilities is 1 both for lines and columns!

Depending on the structure of the transition matrix p (Y / X) the data transmission channels can also be:

Deterministic (in terms of the set X, i.e. the input), if each line of the transition matrix contains one element with probability 1, the remaining elements from the lines being nul (with zero probability);

Quiet/noiseless (in terms of the set Y, i.e. the receiver), if each column of the transition matrix contains one element with non-zero probability, the remaining elements having zero probability!

If the output of a data transmission channel statistically depends on the current input, as well as on the previous inputs and outputs, then the channel is a channel with memory!

It should be noted that this is far from being the only suggestive method of classifying the data transmission channels.

Any data transmission requires a data encryption, for a secure transmission as well as for the need to maximize the relation signal/noise (S/N or SNR). In order to maximize the relation S/N or to restore the useful signal (correction of false bits) we need error-correcting codes, which the more complex they are, the more they hinder the data processing and reception process. For a lower rate of errors and for a shorter period of data processing and reception, we should use a smaller number of bits in encoding the transmitted symbols. An **optimal** encoding!

2. Entropy of a data transmission channel. Information and transinformation of the data channel.

It is usually marked with H(X), where X is the set of events in space of n dimensions: $X = \{x_1, x_2, ..., x_n\}$.

Entropy is measured in bits/events and is given by the relation: $$H(X) = \sum_{x_k} p(x_k) \cdot I(x_k) = \sum_{x_k} p(x_k) \cdot \log_2 \frac{1}{p(x_k)} \quad (4)$$

where k =1, ..., n; and with $I(x_k)$ we mark the information on the event x_k; we notice that the entropy supplies data on the information of the communication channel. They can't be separated. Entropy as a mathematical function (for a closed system) has a parabolic distribution. In order to demonstrate the assertion, we

consider a set $X = \{x_1, x_2\}$ of two events in a closed system and mark $p(x_1) = \omega$, and $p(x_2) = 1- \omega$; thus, the entropical argument function ω will be:

$$H(\omega) = \omega \cdot \log_2 \frac{1}{\omega} + (1-\omega) \cdot \log_2 \frac{1}{1-\omega} \qquad (5)$$

This function has a chart given by the following figure:

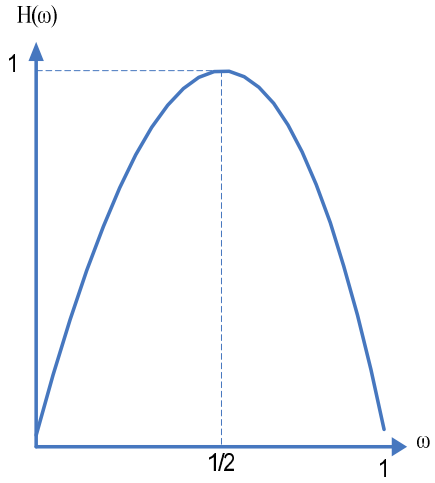

Figure 2. The graphic of Entropy function

Obviously, the function H(ω) has indeterminancy in 0 and 1.
Here are some properties of the entropy:
a) If we have the sets of variables X and Y with identical distributions, then: H(X) = H(Y); but not vice versa!
b) Always (we notice in fig.2) H(X) > 0;
c) There is also a maximum value of entropy as seen in fig.2; the maximum value is obtained when the two events are equiprobable! It can be shown that for a system of n events, the maximum value of entropy is obtained when all the events in the system are equally probable;
d) As a consequence of the previous property, we have: $H(X,Y) \le H(X) + H(Y)$; equality occurs only if X and Y are totally independent from a statistic point of view; the inequality relation may be extended to n sets: $X_1, X_2, ..., X_n$;
e) Also as an outcome of property d) results that *transinformation* defined as I (X, Y) is always higher or equal to zero; equality to zero takes place only if X and Y are totally independent from a statistic point of view; this is a very important property for the data channels because it is mathematically demonstrated that when we send a signal with the transition band B through a data channel characterized by

the relation signal/noise (S/N), then transinformation associated to the signal is given by the relation:

$$I(X,Y) = B \cdot \log_2\left(1+\frac{S}{N}\right) \text{ , bits/second} \qquad (6)$$

3. Definition of optimal encoding and methods of optimal encoding.

We have a code C whose *average code length* $\bar{l}$ is defined as:

$\bar{l} = \sum_{k=1}^{n} p(a_k) \cdot l_k$, relation 2.1; Probability $p(a_k)$ is the probability of the symbol a_k, and l_k is the bit length representation of that symbol; but if any other code C_1 used to encode the same set of data leads to an *average code length* $\overline{l_1}$ that always meets the relation $\bar{l} \le \overline{l_1}$, then the code C is an **optimal code.**

The technical requirements for obtaining an *optimal code*:

I. the least likely primary signals should be encoded through longer code words, and the signals that occur with higher probability should be encoded through shorter code words. Thus, we should respect the set of inequalities:

$$p(a_1) \ge p(a_2) \ge \ldots \ge p(a_n) \text{ and } \overline{l_1} \le \overline{l_2} \le \ldots \le \overline{l_n};$$

II. we should not use the average code length $\bar{l}$ words until we have exhausted all average code length $\overline{l-1}$ words in the encoding operations, otherwise the obtained code will not be *optimal* !

III. It is necessary that the last two code words corresponding to the last two signals with the lowest probabilities have the same code length and differ only through the last binary symbol (bit: one has the binary symbol 0, and the other 1!)

So far there are two methods of optimal encoding which consider these requirements (they can be presented in this paper):

- **The encoding method Shannon – Fano;**
- **The encoding method Huffman.**

3.1. Shannon – Fano method.

It is an arborescent/dendritic method, but for an easy encryption it first implies a tabular representation. Symbols (for example, letters) are written in a table in the descending order of their probabilitiesas they appear (for example in a text). Below we illustrate the encoding method Shannon-Fano through a table associated with a set of 8 characters markes as $a_1, a_2, \ldots, a_8$.

Table.1

Primary symbols	$p(a_n)$	Partitions				Code words	Length l_i
a_1	1/4	0	0			00	2
a_2	1/4		1			01	2
a_3	1/8	1	0	0		100	3
a_4	1/8			1		101	3
a_5	1/16		1	0	0	1100	4
a_6	1/16				1	1101	4
a_7	1/16			1	0	1110	4
a_8	1/16				1	1111	4

We assumed that the symbols a_1, a_2 occur with a probability of 1/4 , a_3 and a_4 with a probability of 1/8, and the others with a probability of 1/16, thus the sum of probabilities should be 1. From the partitioning method as shown in the table, result the code words.

Algorithm Shannon–Fano practically consists of the following steps:

- the set of primary letters/symbols in the table is divided into subsets thus the sum of symbol probabilities of the two subsets is the same; for example:

$p(a_1) + p(a_2) = p(a_3) + p(a_4) + p(a_5) + p(a_6) + p(a_7) + p(a_8) = 1/2$; Similarly, the obtained subsets are again divided in subsets so that the sum of probabilities is equal, and the procedure repeats itself until there are no more than an element subsets (as seen in the table).

It is known that, in general, in order to encode 8 symbols we need at least 3 bits of information. To verify if this type of encoding Shannon-Fano presented in the previous example is a good one, we calculate the entropy (a value, which in the information theory, certifies whether the obtained code is optimal or not). The entropy calculation provides the average length $\bar{l}$ (previously shown). In the case given, we have:

$$H(A)=\sum_{k=1}^{n} p(a_k)\cdot\log\frac{1}{p(a_k)}=2\cdot\frac{1}{4}\cdot\log 4+2\cdot\frac{1}{8}\cdot\log 8+4\cdot\frac{1}{16}\cdot\log 16=1+\frac{3}{4}+1=2{,}75 \quad (7)$$

of course, the log function is in base 2; and A represents the set of symbols a_1, a_2, ..., a_8.

It is noted that we obtained an average length $\bar{l}=2{,}75$ less than 3 (the number of bits necessary to encode 8 symbols). Any other coding variant we try, we won't obtain an average code length lower than 2,75, therefore, we can assert that this obtained code is optimal.

Instead of a tabular representation, we can present the encoding method Shannon-Fano through an arborescent graph (a „tree" structure) shown in the following figure. This is sometimes the easier more understandable encoding method.

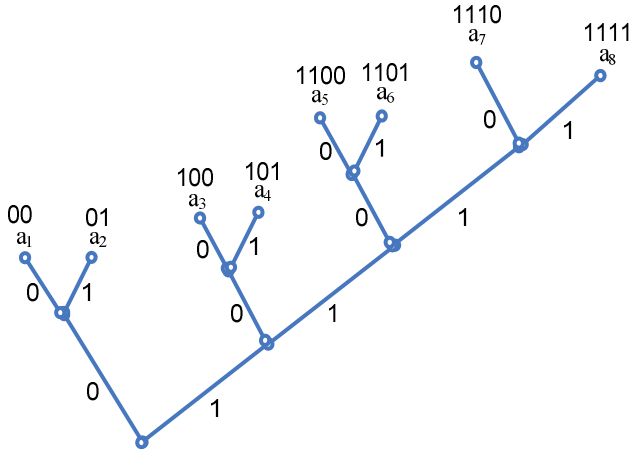

Figure 3. An arborescent graph of the encoding Shannon-Fano method.

The rule is simple: each branch (spatially shifted at about 45 degrees) of the „tree" headed to the left attaches a zero, while each branch headed to the right attaches a 1. We thus obtain the codes for the symbols a_1, a_2, ..., a_8, replacing Table 1. The transit of the „tree" graph is a „down-up" transit, i.e from the „root" to the extremities („leaves").

The Shannon-Fano method is usually applied (as shown in the given example) in the particular cases when the symbol occurence probabilities are integer degrees of 1/2; nevertheless these cases occur often in practice.

Next we will present a method that functions in a series of more general cases, where the symbol occurence probabilities do not have to be integer degrees of 1/2.

3.2. Huffman method.

The Huffman method is based on *reducing the signal source*. We call them *reduced sources*. Let A be a common source of signals represented by the following relation:

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ p(a_1) & p(a_2) & \dots & p(a_n) \end{pmatrix} \tag{8}$$

In the second line of the matrix A the symbol occurence probabilities can be arranged so that they meet the following relation of inequality: $p(a_1) \geq p(a_2) \geq \dots \geq p(a_n)$. After successfully arranging the symbol probabilities in a descending order, we will try to obtain a reduced source A^1 from source A summing up the probabilities two by two and reducing the symbols from two to one. For example:

$$p(a_n^1) = p(a_n) + p(a_{n-1}) \tag{9}$$

the last two symbols become one symbol: a_n^1; thus, the source matrix A type 2 x n becomes a *reduced* source matrix (once) A^1 type 2 x (n-1):

$$A^1 = \begin{pmatrix} a_1 & a_2 & \dots & a_n^1 \\ p(a_1) & p(a_2) & \dots & p(a^1_n) \end{pmatrix} \tag{10}$$

We assume that the reduction can be performed n-N times, thus the sum of probabilities does not exceed value 1. There is also a restrictive condition regarding the reduction: not to spoil the bi-univocity between matrix (source) A and the reduced matrix (reduced source) A^N, a relation which consists in the fact that *if we obtain an optimal code for one of the matrices, then there is always an optimal code for the other one!* ;

Thus, we obtain the final reduced matrix (the final reduced source) A^N, given by the relation: $A^N = \begin{pmatrix} a_1 & a_2 & \dots & a_N \\ p(a_1) & p(a_2) & \dots & p(a_N) \end{pmatrix}$, a matrix 2 x N, where, obviously N < n; and it is desirable for N to be lower than n in order to greatly ease the encoding.

As in the method Shannon – Fano, we present a suggestive encoding example through the source reduction (three times) in a table; we use the initial number of 5 symbols (a_1, a_2, ..., a_5):

Table 2

Initial alphabet			**Reduced sources**					
			Reduced source A^1		**Reduced source A^2**		**Reduced source A^3**	
Primary symbols	**$p(a_k)$**	**Code C**	**$p(a_k^1)$**	**C^1**	**$p(a_k^2)$**	**C^2**	**$p(a_k^3)$**	**C^3**
a_1	0,3	00	0,3	00	**0,45**	1	**0,55**	0
a_2	0,25	01	0,25	01	0,3	00	0,45	1
a_3	0,25	10	**0,25**	1**0**	0,25	01		
a_4	**0,1**	11**0**	**0,2**	1**1**				
a_5	**0,1**	11**1**						

Initially, we start by grouping the lowest symbol occurence probabilities (those bold 0,1 in the table) after previously arranging them in a descending order in column 2 of the table. After obtaining an occurence probability of 0,2, we group again the last smallest probabilities (0,2 with 0,25 ... the bold ones in the table). We notice that, after the first reduction, disappears the last bit of the three from

the last two codes (bolded in the table); thus we reduced the number of bits necessary for the encoding to 2!

After each selection, we always arrange the probabilities in a descending order in the table columns, from top to bottom. After the second reduction disappears again the last bit from group 10 and 11, the symbol a_3^2 being encoded only through one bit with the value 1! The symbols a_1 and a_2 remain encoded through the codes 00 and 01 (we represented them in the table on the lines 2 and 3; they „descended" one line). These will be grouped having the lowest occurence probabilities: 0,3 and 0,25 (we did not bold them in the table!) As 0,3 + 0,25 = 0,55 > 0,45 at the third reduction it will end up on the first line in the table. We cannot have more than 3 reductions, because we obtained only 2 symbols and their occurrence probabilities verify the mathematical rule: 0,45 + 0, 55 = 1! These 2 symbols are encoded only through one bit with the value 0 or 1! We succeeded in reducing the number of bits necessary to the initial encoding (by obtaining reduced sources) from 3 to 1! The obtained code is optimal! Another code with a lower average length does not even exist!

As the Huffman method is an „arborescent" one, we represent the previous example through an arborescent structure:

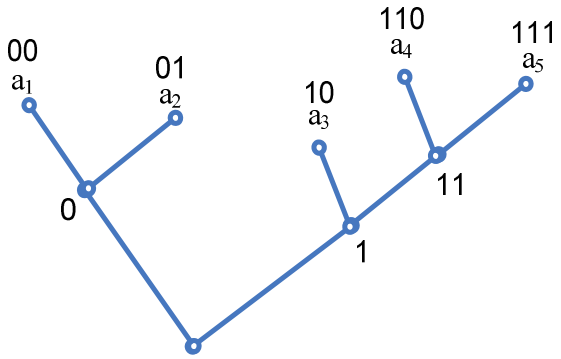

Figure 4. An arborescent structure of Huffman method.

It is noted that this time in the tree nodes we do not have the same number of bits as in the Shannon-Fano method, the number of bits decreasing from the „leaves" towards the „root".

4. Conclusions

The methods of encoding the data in the communication channels increased in the last two decades.

Both presented methods of encoding data in the transmission channels are widely used in the data compression technique.

Eliminating redundant data is very important when dealing with great data flows, transmissions at higher speed and with increased interception safety in the

communication channels. Data compression is used in the new data transmission technologies ATM (Asynchronnous Transfer Mode), SDH, SONET (optical fibre networks), WiMAX , LMDS, etc. and not only.

As concrete applications of the two data compression methods we mention the CD burn (especially the audio data) text compression, archive programs used in PC such as: **.zip** , **.arj** (these are well-known), but not only.

The Huffman encoding is used in the compression of digital images, when each pixel takes values from a finite set (in the case of selfcoloured images from 0 to 255).

References

[1]. Anghel C.V., *Data Security in Wireless Network,* ACTA TECHNICA CORVINIENSIS, BULLETIN of ENGINEERING, Fac. of Engineering Hunedoara, Fasc. 4, 2010;

[2]. Anghel C.V., Petropoulos G., Vaxevanidis N., Dasic P., *Statistical modeling of basic machinability parameters in drilling of metals,* The 19th International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Next Generation of Intelligent Systems and Solutions" 22-25 oct. Trnava, Slovenia, 2008;

[3]. Chioncel C., *Prelucrarea numerică a semnalelor,* Editura Eftimie Murgu, Reşiţa, 2009;

[4]. Răduca E., Răduca M., Ungureanu-Anghel D., *Circuite digitale,* Editura Eftimie Murgu, Reşiţa, 2010;

[5]. Cullman G., *Coduri detectoare şi corectoare de erori,* Editura Tehnică, 1972.

Addresses:

- Drd. Eng. Silviu Drăghici, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, s.draghici@uem.ro
- Lect. Dr. Eng. Cornelia Anghel Drugărin, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, c.anghel@uem.ro
- Lect. Dr. Eng. Cristian Chioncel, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, c.chioncel@uem.ro