POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Communication and Network Engineering

Tesi di Laurea Magistrale

Shannon's theory and some applications

Relatore:
Prof. Camerlo Riccardo

Candidato:
Chen Zhe

Marzo 2018

Abstract

In the late 1940s, Claude Shannon, a research mathematician at Bell Telephone Laboratories, invented a mathematical theory of communication that gave the first systematic framework in which to optimally design communication systems. The main questions motivating this were how to design communication systems to carry the maximum amount of information and how to correct for distortions on the lines. In relative, the contribution of Shannon theory is introduced the concept of information theory and information entropy, where defined a quantity of information.

Shannon's ground-breaking approach introduced a simple abstraction of human communication, called the channel. The communication channel consisted of a transmitter (a source of information), a transmission medium (with noise and distortion), and a receiver (whose goal is to reconstruct the sender's messages).

Information entropy is the most important feature of Shannon theory, which in order to quantitatively analyze transmission through the channel. It introduced a measure of the average quantity of information in a message or event. In general, the more uncertain or random the message is, the more information it will contain.

To complete the quantitative analysis of the communication channel, Shannon introduced the entropy rate, a quantity that measured a source information production rate, and also a measure of the information carrying capacity, called the communication channel capacity.

In information theory, the Shannon–Hartley theorem tells the maximum entropy rate at which information can be transmitted over a communications channel of a specified bandwidth in the presence of noise. In other word, Shannon's development of information theory provided the next big step in understanding how much information could be reliably communicated through noisy channels. Building on Hartley's foundation, Shannon's noisy channel coding theorem (1948) describes the maximum possible efficiency of error-correcting methods versus levels of noise interference and data corruption.

Shannon's theorem shows how to compute a channel capacity from a statistical description of a channel. Given a noisy channel capacity and information transmitted at entropy rate, if entropy rate exceeds the channel capacity, there were unavoidable and uncorrectable errors in the transmission. In convert, there exists a coding technique which allows the probability of error at the receiver to be made arbitrarily small. This means that theoretically, it is possible to transmit information nearly without error up to nearly a limit of channel capacity bits per second.

As for the application, the initial motivation of Shannon theory is to remove the noise during communication, which gives the upper limit of the communication rate. This conclusion was firstly applied on the phone, and later applied on fiber, and now applied on the wireless communication. Today we are able to clearly take ocean telephones or satellite phones, which are closely related to the improvement of communication channel quality. Then, applications extend to biology and chemistry region, like genetic coding.

Contents

Chapter 1

1 Shannon theory

The most important contribution of Shannon theory is introduced the concept of information theory and information entropy, where defined a quantity of information.

1.1 Information theory

Information entropy is the most important feature of Shannon theory, which in order to quantitatively analyze transmission through the channel. It introduced a measure of the average quantity of information in a message or event. In general, the more uncertain or random the message is, the more information it will contain.

1.1.1 Information entropy: a measure for information

The **quantity of information** associated with an event x which has probability π of occurrence is **defined as**

$$I(x) = \log_2 \frac{1}{\pi} = -\log_2 \pi$$

The unit of measurement of the quantity of information is the **information bit**.

More uncertain events have more quantity of information; less uncertain events have less quantity of information; sure events (those with probability $\pi = 1$) have a quantity of information equal to zero. Note that, being $\pi \in [0, 1]$, $I(x) \in [0, \infty]$ which is to say that the quantity of information is always positive or null.

Consider two events x and y, **statistically independent** and with probabilities $\mathrm{p}(x) = \pi_x$ and $\mathrm{p}(y) = \pi_y$. Then the probability that the two events occur at the same time (**joint** probability) is $\mathrm{p}(x, y) = \pi_x \pi_y$, and the quantity of information associated with the occurrence of x and y at the same time is

$$I(x, y) = \log_2 \frac{1}{\mathrm{p}(x, y)} = \log_2 \frac{1}{\pi_x \pi_y} = \log_2 \frac{1}{\pi_x} + \log_2 \frac{1}{\pi_y} = I(x) + I(y)$$

Then the quantity of information of the joint occurrence of two statistically independent events is equal to the sum of their quantities of information.

A communication system is essentially made of an **information source** which produces text, image, music, voice, and so on, and a **receiver** (or **sink**) who wants to read the text, watch the images or listen to the music or voice. Between the source and the sink, there is the communication channel (whose structure will be described in this chapter). Note that there are two basic transmission systems: transmission between two points which are distant **in space** (typical for telephone, television, radio systems, some internet games played through a play-station or similar consoles, etc.) or **in time** (typical for storage systems like music or films stored in a CD or DVD disk, or text files or images in the computer hard disk, etc.). These two transmission systems have some peculiar differences so that a system which is suitable for a telephone system may not be suitable for file storage. The Shannon theory would be mainly denoted to the case of transmission between two distant points in space (e.g. phone system).

1.1.1.1 Discrete source

Consider then the simple case of a source which generates just one symbol x, randomly choosing it in the set $\mathrm{X} = \{x_1, x_2, \dots, x_N\}$ (the **source alphabet**). In terms of probability theory, x is a discrete random variable.

Let $\mathrm{P}(x = x_k) = \pi_k$ be the probability that the source chooses symbol x_k. Since the events $x = x_k$ and $x = x_i$ are all mutually exclusive for $i \neq k$ (it is impossible that the source generates both x_k and x_i at the same time), and for sure the source generates **a** symbol, then we must have (second[1] and third[2] axioms of probability theory):

$$\sum_{k=1}^{N} \pi_k = 1$$

The quantity of information associated with the event "the source generates symbol x"is

$$I(x) = \log_2 \frac{1}{P(x)}$$

and $I(x)$ is a random variable, which takes value $I(x = x_k) = -\log_2 \pi_k$ with probability π_k; you can image that a nonlinear system exists such that, when the input is $x = x_k$,the output is relatively as $I(x = x_k) = -\log_2 \pi_k$, then $I(x)$ is a random variable obtained from nonlinear transformation of the input random variable x.

The **average quantity of information** associated with the generation of one symbol is

[1] This is the assumption of unit measure: that the probability that at least one of the elementary events in the entire sample space will occur is 1. $\mathrm{P}(\Omega) = 1$

[2] This is the assumption of σ– additibity: Any countable sequence of disjoint sets (synonymous with mutually exclusive events) $E_1, E_2, \dots$ satisfies: $\mathrm{P}(\bigcup_{i=1}^{\infty} E_i) = 1$

$$H(x) = E\{I(x)\} = \sum_{k=1}^{N} \pi_k I(x = x_k) \; = -\sum_{k=1}^{N} \pi_k \log_2 \pi_k$$

The average quantity of information $H(x)$ produced by the source each time it generates a symbol is called **source entropy**. Being the average of nonnegative random variable $I(x)$, the entropy is nonnegative (it can be zero).

THEORM:
The maximum value of entropy for a source with N symbols is $\log_2 N$ and it is obtained when the N symbols x_k are all equally likely (i.e. $\pi_k = 1/N$).

PROOF:

$$\begin{aligned}
H(x) - \log_2 N &= -\sum_{k=1}^{N} \pi_k \log_2 \pi_k \; - \left(\sum_{k=1}^{N} \pi_k\right) \log_2 N \\
&= -\sum_{k=1}^{N} \pi_k \log_2 \pi_k \; - \left(\sum_{k=1}^{N} \pi_k \log_2 N\right) \\
&= -\sum_{k=1}^{N} \pi_k \left(\log_2 \pi_k \; + \log_2 N\right) = \sum_{k=1}^{N} \pi_k \log_2 \frac{1}{\pi_k N} \\
&= \frac{1}{\ln 2} \sum_{k=1}^{N} \pi_k \left(\ln \frac{1}{\pi_k N}\right) \le \frac{1}{\ln 2} \sum_{k=1}^{N} \pi_k \left(\frac{1}{\pi_k N} - 1\right) \\
&= \frac{1}{\ln 2} \sum_{k=1}^{N} \frac{1}{N} - \frac{1}{ln\,2} \sum_{k=1}^{N} \pi_k = \frac{1}{\ln 2}\left(N \cdot \frac{1}{N} - 1\right) = \frac{1}{\ln 2}(1 - 1) = 0
\end{aligned}$$

The key point in the above proof is that

$$\ln x \le x - 1 \quad \forall\, x > 0$$

In brief, we can say that the **discrete entropy is maximized by the uniform discrete probability density function.**

One useful inequality is the following:

$$\sum_{k=1}^{N} \pi_k \log_2 \pi_k \; \ge \sum_{k=1}^{N} \pi_k \log_2 \rho_k$$

The proof is similar to the previous one:

$$\sum_{k=1}^{N}\pi_k\log_2\rho_k-\sum_{k=1}^{N}\pi_k\log_2\pi_k=\sum_{k=1}^{N}\pi_k\log_2\frac{\rho_k}{\pi_k}=\frac{1}{\ln 2}\sum_{k=1}^{N}\pi_k\ln\frac{\rho_k}{\pi_k}$$

$$\leq\frac{1}{\ln 2}\sum_{k=1}^{N}\pi_k\left(\frac{\rho_k}{\pi_k}-1\right)=\frac{1}{\ln 2}\sum_{k=1}^{N}(\rho_k-\pi_k)=\frac{1}{\ln 2}\left(\sum_{k=1}^{N}\rho_k-\sum_{k=1}^{N}\pi_k\right)$$

$$=\frac{1}{\ln 2}(1-1)=0$$

The quantity $\sum_{k=1}^{N}\pi_k\log_2\frac{\pi_k}{\rho_k}$ is called **relative entropy**, and we just showed it is always non-negative.

Going back to entropy $H(X)$, for the simple case $N=2$, which occurs for example when the source generates ether “0” or “1”, the entropy of binary random variable with probability vector $(p,1-p)$ is

$$H(X)=-p\log_2 p-(1-p)\log_2(1-p)\triangleq\mathcal{H}(p)$$

Which is plotted in figure 1.1: the maximum of the function is 1 at $p=0.5$, whereas $H(X)=0$ when $p=0$ or $p=1$ (in the last case, the source always generates x_1 or always generate x_2, thus producing no information at all).

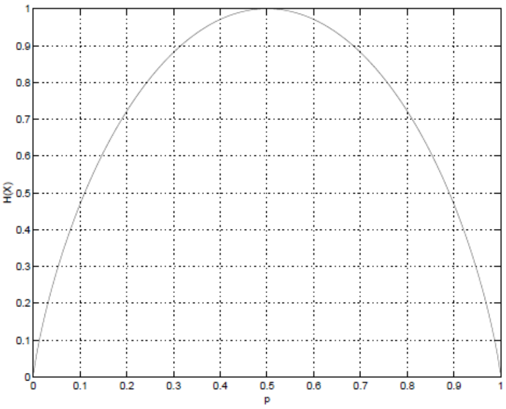


Figure1.1: Entropy of a binary source which generates symbol x_1 with probability p and symbol x_2 with probability $1-p$.

If $p=1/4$, for instance, then symbol x_1 carries $\log_2 4=2$ information bits, while x_2 carries $\log_2(4/3)=0.415$ information bits; in the average the information quantity of source is

$$H(X)=\frac{1}{4}\times 2+\frac{3}{4}\times 0.415=0.811\ [information\ bits]$$

The source entropy can be seen as the average quantity of information necessary to know which symbol has been generated by the source. Consider the following “game”: a person A observes the symbol generated by the source, whereas as a second person B must ask questions to A in order to know the value of x. The game is such that B

can only ask questions that admit either answer “yes” or answer “no”. It is as if as A becomes a new source of information endowed with the two elements alphabet {“yes”, “no”}, and we have seen that the maximum entropy of A is 1 bit, when the two symbols are equally likely; then B gets the maximum information (1 bit), if it has question is such that the answer is “yes” with probability 0.5. Now suppose that the original source generate symbol $\{x_1, x_2, x_3, x_4\}$ with probabilities: $\pi_1 = 1/2, \pi_2 = 1/4$ and $\pi_3 = \pi_4 = 1/8$.

- If B is smart, he starts asking if $x = x_1$, and A answers “yes” with probability $\pi_1 = 1/2$, so that B wins the game with just 1 question. With this smart question, B gets exactly 1 bit of information from A.
- If A answers “no” to the first question, then B asks if $x = x_2$, and A answers “yes” with probability $\frac{\pi_2}{\pi_2+\pi_3+\pi_4} = \frac{1}{2}$ (this is the probability that the source generates x_2, knowing that it has not generate x_1). Also with this second question B gets exactly one bit of information from A. If A answers “yes” to the second question then the game ends with two questions (and 2 bits of information provided from A). The probability that B need two questions to know the value of x is $\pi_2 = 1/4$(the probability that the source has generated x_2).
- If A answers “no” to the second question, then B asks if $x = x_3$, and A answers “yes” with probability $\frac{\pi_3}{\pi_3+\pi_4} = \frac{1}{2}$; if A answers “no” to the third question, then B knows that the generated symbol is x_4 and it is not necessary to ask other questions.

Then the game ends with three questions (and 3 information bits provided by A). The probability that B needs three questions to know the value of x, then B needs an average number of questions equal to

$$\overline{n} = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{4} = \frac{7}{4}$$

Noticed that the source entropy is

$$H(X) = -\sum_{i=1}^{N} \pi_i \log_2 \pi_i \; = \frac{7}{4}$$

Which means that, in this case, the average number of questions exactly equal to the source entropy. Notice that how much $\overline{n}$ is close to $H(X)$ depends on how smart is our friend B.

If the symbols generated by the source are equally likely, then B has the maximum uncertainty on the value of x; if the source generates always the same symbol $x = x_1$, then B directly knows the value of x (without any question); if the symbols are not equally likely (intermediate case between the previous two), then B knows that one symbol is more probable and can use this knowledge to devise a smart strategy to reduce the number of questions.

On the basis of the above game, we can state that $H(X)$ is the quantity of information that, in the average, must be provided to the receiver so that it can almost surely know (i.e. with error probability equal to zero) the symbol generated by the source. It is the task of transmission system, made of the modulator, channel, the demodulator, to give the detector inside the receiver the maximum possible quantity of information.

In real life the source does not generate one symbol, but it continuously outputs symbols, one every symbol duration T_s seconds. We talk about a **memoryless source**, if it does not build new symbol using previously generated symbols, or better, if the symbols generated by the source are statistically independent random variable taking values in the set $X = \{x_1, x_2 \dots x_M\}$. Then the sequence of two consecutive symbols $\left[x^{(n)}, x^{(n+1)}\right]$ takes values in the (ordered) set of values $X^2 = \{[x_1, x_1], \dots [x_1, x_M], [x_2, x_1], [x_2, x_2], \dots [x_2, x_M], \dots [x_M, x_M]\}$, with M^2 symbols. The source is further divided into two categories, including memoryless and memory:

- *Memoryless source:*

If the source is memoryless, then

$$P\left(x^{(n)} = x_k, x^{(n+1)} = x_i\right) = P\left(x^{(n)} = x_k\right) P\left(x^{(n+1)} = x_i\right) = \pi_k \pi_i$$

And the quantity of information associated with the event $\{x^{(n)} = x_k, x^{(n+1)} = x_i\}$ is the sum of the quantities of information associated with the two sample event $\{x^{(n)} = x_k\}$ and$\{x^{(n+1)} = x_i\}$, which is to say that:

$$\mathrm{I}\left(\{x^{(n)} = x_k, x^{(n+1)} = x_i\}\right) = \mathrm{I}\left(\{x^{(n)} = x_k\}\right) + \mathrm{I}\left(\{x^{(n+1)} = x_i\}\right)$$

$$= -\log_2 \pi_k - \log_2 \pi_i$$

Then the average quantity of information of two subsequent symbols is

$$H(X^2) = E\left\{\mathrm{I}\left(\{x^{(n)}, x^{(n+1)}\}\right)\right\} = E\left\{I\left(x^{(n)}\right) + I\left(x^{(n+1)}\right)\right\}$$

$$= E\left\{I\left(x^{(n)}\right)\right\} + E\left\{I\left(x^{(n+1)}\right)\right\} = 2H(X)$$

Notice that memoryless source generates each symbol in the same way. I.e. the $(n+1)_{\mathrm{th}}$ symbol has the same entropy as the n_{th} symbol.

Therefore, extending this result, if the source is **memoryless**, we get that

$$\boldsymbol{H(X^N) = NH(X)}$$

I.e. the entropy of N symbols is N times the entropy of one symbol.

- *Memory source:*

If the source has **memory**, then

$$\boldsymbol{H(X^N) < NH(X)}$$

The proof is similar to the memoryless case:
If the source has memory, then the symbol generation depends on the previous generated symbol. The probability of generation follows the Bayes' Rule:

$$P\left(x^{(n)} = x_k, x^{(n+1)} = x_i\ \right) = P\left(x^{(n+1)} = x_i\ |x^{(n)} = x_k\right)P\left(x^{(n)} = x_k\right)$$

And the quantity of information associated with the event $\{x^{(n)} = x_k, x^{(n+1)} = x_i\ \}$ is the sum of the quantities of information associated with the two sample event $\{x^{(n+1)} = x_i\ |x^{(n)} = x_k\}$ and $\{x^{(n)} = x_k\}$, which is to say that:

$$\mathrm{I}\left(\{x^{(n)} = x_k, x^{(n+1)} = x_i\}\ \right) = \mathrm{I}\left(\{x^{(n+1)} = x_i\ |x^{(n)} = x_k\ \}\right) + \mathrm{I}\left(\{x^{(n)} = x_k\}\right)$$

Then the average quantity of information of two subsequent symbols is

$$H(X^2) = E\left\{\mathrm{I}\left(\{x^{(n)}, x^{(n+1)}\}\right)\right\} = E\left\{\mathrm{I}\left(\{x^{(n+1)}\ |x^{(n)}\}\right) + I\left(x^{(n)}\right)\right\}$$

$$= E\left\{\mathrm{I}\left(\{x^{(n+1)}\ |x^{(n)}\}\right)\right\} + E\left\{I\left(x^{(n)}\right)\right\} = H(X_{n+1}|X_n) + H(X_n)$$

$$< H(X_{n+1}) + H(X_n) = 2H(X)$$

For the memory source, the key point of proof is

$$H(X_{n+1}|X_n) < H(X_{n+1})$$

since observation of next symbol carries some information based on previous symbol and therefore the uncertainty next symbol is reduced. Extending the result, we get that

$$H(X^N) < NH(X)$$

As an example, consider the Italian language, for which letter *"q"* is almost always followed by *"u"*. Assume to repeat the same game of "guessing the symbol" described above, which, in this case, becomes "guessing the sentence". Assume that A and B are Italian guys and A thinks of the word *"questo"*, and B has to guess this word. Then B starts with the first letter of the word and, after some questions and answers, knows that the first letter is *"q"*. Then B has much less uncertainty on the value of second letter and start asking about the third letter, which, almost certainly, is a vowel (only 5 possibilities). This example shows that, while the first letter of the word has an entropy equals to $\log_2 26$ (being 26 the number of letters in alphabet), the second one has a much smaller entropy (it is not exactly *zero*, because A could think of a strange word, maybe an acronym).

1.1.1.2 Analog source

In some cases, the source generates a continuous random variable. For example, this occurs if we consider a voice random process sampled at 8000 hertz: each sample is a random variable, which is typically considered as a bilateral exponential (or Laplace) random variable. Let $f_x(u)$ be the probability density function of the continuous random variable x; remember that

$$\mathrm{P(a \leq x \leq b)} = \int_a^b f_x(u)\,du$$

Then, extending the definition of discrete entropy, we say that the **differential entropy** of the continuous source is

$$h(x) = \int_{-\infty}^{\infty} f_x(u)\log_2 \frac{1}{f_x(u)}\,du = -\int_{-\infty}^{\infty} f_x(u)\log_2 f_x(u)\,du$$

1.1.1.2.1 Maximum differential entropy

THEOREM:

For a given mean μ_x and variation σ_x^2, whose definition are

$\mu_x = \int_{-\infty}^{\infty} u\, f_x(u)\,du$, $\sigma_x^2 = \int_{-\infty}^{\infty} (u-\mu_x)^2 f_x(u)\,du$

the probability density function that maximizes the differentially entropy is the Gaussian one:

$$f_x(u) = \frac{1}{\sqrt{2\pi\sigma_x^2}}\exp\left\{-\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\}$$

Relatively, the maximum differential entropy is

$$\boldsymbol{h(x) = \frac{1}{2}\log_2(2\pi e\sigma_x^2)}$$

For the Gaussian probability density function, the maximum differential entropy can be evaluated as follows:

$$\begin{aligned}
h(x) &= \int_{-\infty}^{\infty} f_x(u)\log_2 \frac{1}{f_x(u)}\,du \\
&= \int_{-\infty}^{\infty} f_x(u)\left[\log_2\sqrt{2\pi\sigma_x^2} + \log_2 \exp\left\{\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\}\right]du \\
&= \log_2\sqrt{2\pi\sigma_x^2}\int_{-\infty}^{\infty} f_x(u)\,du + \int_{-\infty}^{\infty} f_x(u)\left\{\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\}(\log_2 e)du \\
&= \log_2\sqrt{2\pi\sigma_x^2} + \int_{-\infty}^{\infty} f_x(u)\left\{\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\}(\log_2 e)du \\
&= \log_2\sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2}\int_{-\infty}^{\infty} f_x(u)\,(u-\mu_x)^2\,du \\
&= \log_2\sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2}\sigma_x^2 = \frac{1}{2}\log_2 2\pi\sigma_x^2 + \frac{1}{2}\log_2 e \\
&= \frac{1}{2}\log_2(2\pi e\sigma_x^2)
\end{aligned}$$

It is interesting to note that the result does not depend on the mean μ_x, but only of the variance σ_x^2: the mean does not carry information, while the information is “stored” in the variance of x. The mean is predictable, how much x is far from its mean is the real information. The fact that $h(x)$ does not depend on μ_x is valid for any probability density function, even if it was shown just for the Gaussian probability density function.

Proof:

- We denoted by y and x the random variable with joint probability density functions $f_y(u)$ and $f_x(u)$. Let $f_y(u)$ and $f_x(u)$ be respectively, a general probability density function and a Gaussian random variable probability density function, corresponding to the same variance, i.e.
$$\sigma_y^2 = \sigma_x^2$$
Since the mean value does not affect the entropy, we assume that in both cases the mean value is zero,
$$\mu_y = \mu_x = 0$$
- Note that if the differential entropy is
$$h(x) = \int_{-\infty}^{\infty} f_x(u) \log_2 \frac{1}{f_x(u)} du = -\int_{-\infty}^{\infty} f_x(u) \log_2 f_x(u)\, du$$
for events with continuous support where
$$\int_{-\infty}^{\infty} f_x(u)\, du = 1$$
Extending this definition, we get
$$h(y) = \int_{-\infty}^{\infty} f_y(u) \log_2 \frac{1}{f_y(u)} du = -\int_{-\infty}^{\infty} f_y(u) \log_2 f_y(u)\, du$$
$$\int_{-\infty}^{\infty} f_x(u)\, du = \int_{-\infty}^{\infty} f_y(u)\, du = 1$$
- First of proof, note that
$$\begin{aligned}\int_{-\infty}^{\infty} f_y(u) \log_2 \frac{1}{f_x(u)} du &= \int_{-\infty}^{\infty} f_y(u) \left[\log_2 \sqrt{2\pi\sigma_x^2} + \log_2 \exp\left\{\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\}\right] du \\ &= \log_2 \sqrt{2\pi\sigma_x^2} \int_{-\infty}^{\infty} f_y(u)\, du + \int_{-\infty}^{\infty} f_y(u) \left\{\frac{(u-\mu_x)^2}{2\sigma_x^2}\right\} (\log_2 e) du \\ &= \log_2 \sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2} \int_{-\infty}^{\infty} f_y(u)\, (u-\mu_x)^2\, du \\ &= \log_2 \sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2} \int_{-\infty}^{\infty} f_y(u)\, (u-\mu_y)^2\, du \\ &= \log_2 \sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2} \sigma_y^2 = \log_2 \sqrt{2\pi\sigma_x^2} + \frac{(\log_2 e)}{2\sigma_x^2} \sigma_x^2 \\ &= \frac{1}{2} \log_2 2\pi\sigma_x^2 + \frac{1}{2} \log_2 e = \frac{1}{2} \log_2 (2\pi e \sigma_x^2) = h(x) \\ &= \int_{-\infty}^{\infty} f_x(u) \log_2 \frac{1}{f_x(u)} du\end{aligned}$$
Then we get
$$h(x) = \int_{-\infty}^{\infty} f_x(u) \log_2 \frac{1}{f_x(u)} du = \int_{-\infty}^{\infty} f_y(u) \log_2 \frac{1}{f_x(u)} du$$

- Next we calculate the difference and apply the inequality $\ln x \leq x-1$ holding for every $x > 0$:

$$\begin{aligned}
h(y)-h(x) &= \int_{-\infty}^{\infty} f_y(u)\log_2 \frac{1}{f_y(u)}\,du - \int_{-\infty}^{\infty} f_x(u)\log_2 \frac{1}{f_x(u)}\,du \\
&= \int_{-\infty}^{\infty} f_y(u)\log_2 \frac{1}{f_y(u)}\,du - \int_{-\infty}^{\infty} f_y(u)\log_2 \frac{1}{f_x(u)}\,du \\
&= \int_{-\infty}^{\infty} f_y(u)\log_2 \frac{f_x(u)}{f_y(u)}\,du = \int_{-\infty}^{\infty} f_y(u)\ln\left(\frac{f_x(u)}{f_y(u)}\right) du \log_2 e \\
&\leq \int_{-\infty}^{\infty} f_y(u)\left(\frac{f_x(u)}{f_y(u)}-1\right) du \log_2 e \\
&= \int_{-\infty}^{\infty} \left(f_x(u)-f_y(u)\right) du \log_2 e \\
&= \left\{\int_{-\infty}^{\infty} f_x(u)\,du - \int_{-\infty}^{\infty} f_y(u)\,du\right\}\log_2 e = (1-1)\log_2 e = 0
\end{aligned}$$

Obviously, this is an extension of corresponding inequality:

$$h(y) \leq h(x)$$

which holds provided that y and x have the same variance and x is a Gaussian random variable.

1.1.1.2.1 Negative entropy

The differential entropy can be negative. Thus the differential entropy loses the natural property of entropy of being positive.

For examples:

- *Uniform distribution*

Consider a random variable distributed uniformly from *0* to a, so that its probability density function (uniform pdf[3]) is $\frac{1}{a}$ from *0* to a and *0* elsewhere. Then its differential entropy is

$$h(x) = -\int_{-\infty}^{\infty} f_x(u)\log_2 f_x(u)\,du = -\int_0^a \frac{1}{a}\log_2\frac{1}{a}\,du = \log_2 a$$

Note: for $a < 1$, $\log_2 a < 0$, and the differential entropy is negative. If $a = \frac{1}{2}$, the uniform pdf in [0, 1/2], we have

[3] In probability theory and statistics, the continuous uniform distribution or rectangular distribution is a family of symmetric probability distributions such that for each member of the family, all intervals of the same length on the distribution's support are equally probable. The support is defined by the two parameters, a and b, which are its minimum and maximum values. The distribution is often abbreviated U(a,b). The probability density function of the continuous uniform distribution is $\mathrm{f(x)} = \begin{cases} \frac{1}{b-a}, & \mathrm{a < x < b} \\ 0, & \mathrm{otherwise} \end{cases}$. The information quantity is $\mathrm{I(x)} = \log_2 \frac{1}{b-a}$.

$$h(x) = -\int_{-\infty}^{\infty} f_x(u)\log_2 f_x(u)\,du = -\int_0^{\frac{1}{2}} 2\log_2 2du = -\int_0^{\frac{1}{2}} 2du = -2 \times \frac{1}{2} = -1$$

Hence, unlike discrete entropy, differential entropy can be negative.

- *Gaussian distribution*

The entropy of the Gaussian density on $\boldsymbol{R}$ with mean μ and variance σ^2 is

$$-\int_{\boldsymbol{R}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left(-\log_2\sqrt{2\pi\sigma^2} - \frac{(x-\mu)^2}{2\sigma^2}\right)dx = \frac{1}{2}(1+\log_2(2\pi\sigma^2)).$$

The mean μ does not enter the final formula, so all Gaussian with a common σ^2 have the same entropy.

For σ near 0, the entropy of a Gaussian is negative. Graphically, when σ is small, a substantial piece of the probability density function has values greater than 1, and there entropy $-p\log_2 p < 0$, where p as the probability of one event. For discrete distributions, entropy is always positive, since values of a discrete probability function never exceed 1.

- *Exponential distribution*

The entropy of the exponential density on $(0, \infty)$ with mean λ is

$$-\int_0^{\infty} \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \left(-\log_2 \lambda - \frac{x}{\lambda}\right) dx = 1 + \log_2 \lambda$$

As in the previous example, this entropy becomes negative for small λ.

Entropy is the amount of disorder that is in a system. Disorder is much more probable to occur than order in a random system. Negative entropy is reverse entropy. So if entropy is the amount of disorder, negative entropy means something has less disorder, or more order. In other words, negative entropy is the measure of "order", which means organization, structure and function: the opposite of randomness or chaos, in system.

The negative entropy has different meanings in information theory, thermodynamics and theoretical biology.

- *Information theory*

On the one hand, in information theory and statistics, negative entropy is used as a measure of distance to normality. Since out of all distributions with a given mean and variance, the normal or Gaussian distribution is the one with the maximum entropy (proof in section 1.1.1.2.1). Negative entropy measures the difference in entropy between a given distribution and the Gaussian distribution with the same mean and variance.

On the other hand, the information can be seen as negative entropy. In general, information can be defined as: information is the essential characteristics of the physical system movement, including the way, the state and the order of movement. And entropy is simply understood as disorder state in a system of material movement, so the negative entropy is an orderly state. For instances, as for learning, negative entropy can be transformed into the cerebral cortex of information; as for motion, negative entropy can be transformed into the muscle memory. The information is passed through the specific form of data being processed, and the information has an added value that exceeds the value of the data itself, which has the similar definition of the negative entropy. Therefore, we can think that the information is negative entropy.

- *Thermodynamics*

According to the second law of thermodynamics when we look at the system as whole entropy will always increase, as negative entropy must be balanced out by, most likely more, positive entropy. Negative entropy can only occur when we look at a small portion of the world. For instance, if you need to fold your shirt you are using energy and thus you are becoming more disordered. The shirt is now less disordered but you are more disordered and thus the system as a whole is in a state of either zero entropy or positive entropy although the shirt itself is in a state of negative entropy.

We cannot simply look at a single object at a single point in time to determine if it has negative entropy. In order to determine entropy it must be compared either to itself at a previous or later point in time or to something else. For example, thinking about your room at this very instant, the bed may not be made and there may be a shirt on the floor. You may think about this as clean (ordered) or as a mess (disordered). If yesterday the floor was spotless and the bed nicely made then you are moving towards more disorder, so it is in a state of positive entropy. But if yesterday you didn't even have a sheet on the bed and all of your laundry was scattered across the floor then you are now moving towards less disorder so it is in a state of negative entropy.

1.1.2 Source Coding

An information source generates a sequence of symbols, and they carry information. Typically each generated symbol does not carry the maximum possible quantity of information, so that there is space for optimizing the system and reduce the number of generated symbols keeping the same information quantity. The source coder performs the operation of compressing the output of the original source, so that the information can be stored in a smaller file or it can be transmitted in a shorter time or in a smaller bandwidth. The system is then made of: the original source, the source encoder, the transmitter, the receiver, the source decoder, the final user or sink.

There are two main families of source encoder:

1. The **lossless** source coders are those which preserve the quantity of information of the original source: the operation performed by the encoder is invertible and it is always possible to get the input of the encoder from its output;
2. The **lossy** source coders are those which reduce the number of generated symbols with a process which is not exactly invertible; from the output of the encoder it is possible to build a sequence of symbols which “looks like” the true input of the encoder but is not exactly equal to it.

Lossless encoders are used when the source produces text, executable files, etc.; lossy encoders are used for images, music, voice etc. In lossy encoders the similarity between the original symbols of the source and the symbols reconstructed by the source decoder is typically perceptual: a team of experts establishes the quality of compression.

The lossless encoding technique contains the lossless Huffman source encoder; other lossless encoders are the arithmetic encoder, the Shannon encoder, the Lempel-Ziv encoder. The lossy encoders strictly depend on the source (whether itis voice or image, etc.); JPEG, MPEG etc. are examples of lossy encoders.

In general, the source encoder takes an input symbol/sample (or a sequence of input symbols/samples) and outputs a sequence of binary digits (bits). Depending on how the mapping is performed, source encoders are further divided into two categories:

- **fixed length source encoders**: they map m input symbols to the n output bits, where m and n are constant and fixed
- **Variable length source encoders**: they map m input symbols to the n output bits, but n depends on the specific input. For example a variable length encoder might map the two subsequent input symbols x_1x_3 into the sequence $1100101(m = 2$ and $n = 7)$, and the input symbols x_1x_2 into $1011(m = 2$ and $n = 4)$.

The source decoder takes the input bits and generates symbols/samples. In the preliminary analysis of source encoders, it is typical to assume that the transmitter, channel, and receiver are ideal, which allows to consider an equivalent system in which we have just the source, the encoder, the decoder and the destination/sink.

1.1.2.1 A very simple fixed-length source encoder

Assume that you have a discrete source with alphabet $X = \{x_1, x_2, \dots, x_M\}$ and corresponding probabilities$\pi_1, \pi_2, \dots, \pi_M$, **being M an integer power of 2**. If $\pi_1 = \pi_2 = \cdots = \pi_M = 1/M$, then the entropy of source is $\log_2 M$ and each event $x = x_k$ carries the same information quantity $\log_2 M$. The source encoder maps each symbol

x_k to a sequence of bits, and the number of bits required to represent and distinguish the M symbols is $\log_2 M$. The actual mapping is not important. Let us say that x_1 is mapped to the binary representation of number 0 (using $\log_2 M$ bits), x_2 is mapped to the binary representation of number 1 and x_k is mapped to the binary representation of number $k-1$. For example, for $M=4$, we have the mapping shown in table 1.1.

k	symbol	mapping
1	x_1	00
2	x_2	01
3	x_3	10
4	x_4	11

Table 1.1: Mapping between source symbols and their binary encoding, for the case of $M=4$ equally likely symbols.

The important point is that the mapping between symbol and its binary representation (or **code word**) is one-to-one: given the symbol we uniquely know its binary representation, or given the binary representation we uniquely know its corresponding symbol. Note that, once the mapping between symbols and their code words has been decided, the software or hardware that performs the direct or inverse mapping never makes errors. The fact that no errors are made means that all the information generated by the source is correctly received and managed by the source decoder, and this means that the binary code word made of $\log_2 M$ bits actually carries $\log_2 M$ information bits.

Then, the source generates a sequence of symbols, each of which carries $\log_2 M$ information bits, the source encoder outputs $\log_2 M$ bits for each input symbol, the source decoder maps groups of $\log_2 M$ input bits again into the symbol generated by the source, without making errors. If we look at the binary sequence generated by the source encoder, we can think that it is the output of an "equivalent" binary source. Now, if $\log_2 M$ bits at the output of the source encoder actually carry $\log_2 M$ information bits, then in the average each bit carries one information bit[4]. But only a binary source with equally likely symbols have an entropy equals to 1 information bits, which means that the source encoder actually generates bit "1" with probability 1/2 and bit "0" with probability 1/2. Note that this is the best we can do: it is not possible that the "equivalent" binary source at the output of the source encoder can associate more than one information bit to each output bit.

[4] Note the difference between bit, which is the output of the source and randomly takes one of the two values "0" or "1", and the information bit, which is a measure of the information content of that bit.

1.1.2.2 Source coding theorem and Kraft inequality

In information theory, Shannon's source coding theorem (or noiseless coding theorem) establishes the limits to possible data compression, and the operational meaning of the Shannon entropy.

The source coding theorem places an upper and a lower bound on the minimal possible expected length of code words as a function of the entropy of the input word (which is viewed as a random variable) and of the size of the target alphabet.

Kraft inequality is the sufficient and necessary condition to guarantee the code word uniquely decoded.

The source coding theorem (Shannon 1948) shows that (in the limit, as the length of a stream of independent and identically-distributed random variable (*i.i.d.*[5]) data tends to infinity) it is impossible to compress the data such that the code rate (average number of bits per symbol) is less than the Shannon entropy of the source, without it being virtually certain that information will be lost. However it is possible to get the code rate arbitrarily close to the Shannon entropy, with negligible probability of loss.

1.1.2.2.1 Source coding theorem

The previous simple example allows us to understand that the ideal source encoder generates equally likely bits and each of these bits carries exactly one information bit. Moreover, if symbol x_k has a quantity of information $\mathrm{I}(x = x_k)$, then it should be mapped into a number of bits n_k equal to $\mathrm{I}(x = x_k)$ so that each generated bit carries exactly one bit of information. Certainly, this is possible only if $\mathrm{I}(x = x_k)$ is integer; if $\mathrm{I}(x = x_k)$ is not integer, then we can take n_k integer with $n_k > \mathrm{I}(x = x_k)$. Note that we would like to have one information bit for each encoded bit, **in the average**, not for each symbol. This means that the choice $n_k > \mathrm{I}(x = x_k)$, when $\mathrm{I}(x = x_k)$ is not integer might lead to inefficient codes, while the choice $\mathrm{n}_k = \mathrm{I}(x = x_k)$ when $\mathrm{I}(x = x_k)$ is integer for all values of k is optimum.

Let us consider the following case: source with alphabet $X = \{x_1, x_2, \dots, x_M\}$ and associated probabilities $\{\pi_1, \pi_2, \dots, \pi_M\}$, source encoder that maps symbol x_k into a sequence of n_k bits with

$$\mathrm{I}(x = x_k) < n_k < \mathrm{I}(x = x_k) + 1$$

Note that in general $\mathrm{I}(x = x_k) = -\log_2 \pi_k$ is not an integer, while n_k must be an integer; however, it is always possible to find an integer value in the range $[\mathrm{I}(x =$

[5] In probability theory and statistics, a sequence or other collection of random variables is **independent and identically distributed** (i.i.d. or iid or IID) if each random variable has the same probability distribution as the others and all are mutually independent.

$x_k), \mathrm{I}(x = x_k) + 1[$. Define as n the random variable corresponding to the number of bits produced by the source encoder, when its input is the random variable x. Then the mean value of n (**mean length of the source code word**) is

$$\bar{n} = \mathrm{E}\{n\} = \sum_{k=1}^{M} n_k \pi_k$$

And, being $I(x = x_k) \leq n_k < I(x = x_k) + 1$,

$$\sum_{k=1}^{M} I(x = x_k)\,\pi_k \leq \bar{n} < \sum_{k=1}^{M} [I(x = x_k) + 1]\,\pi_k$$

Finally, we get the relationship between mean length of source code word and entropy:

$$H(X) \leq \bar{n} \leq H(X) + 1$$

This last inequality $\boldsymbol{H(X) \leq \bar{n} \leq H(X) + 1}$ is called the **source coding theorem** and gives an upper and a lower bound to the mean length of the source code words.

1.1.2.2.2 Kraft inequality

However, we only proved that, if the source encoder use the n_k as specified, then $H(X) \leq \bar{n} \leq H(X) + 1$, but we did not prove that such a mapping exists and that the obtained code is uniquely denoted.

First of all, it is necessary to give a formal definition to the word "**uniquely decodable**": **a source code is uniquely decodable if none of its code words is prefix of another code word.**

For example, assume that a source encoder uses the following three code words: "01111" for symbol x_1, "01" for symbol x_2 and "111" for symbol x_3. Assume that the decoder has the input sequence "01111": how can the decoder decider whether this bit sequence corresponds to x_1 or to x_2, x_3? Both hypotheses are feasible and there is no way to find out which is the correct one. The code of this example is not uniquely decodable in that code "01" (for symbol x_2) is prefix of code word "01111" (for symbol x_1).

One way to easily check whether a code is uniquely decodable is to arrange its code words into a (horizontal) binary tree like the one shown in Fig. 1.2. Starting from the root, and using the convention that the upper branch corresponds to bit "1" while the lower branch corresponds to bit "0", it is easy to associate a node or a leaf of the binary tree to a code word. Figure 1.2 shows the position of code words "11", "100", "1100", "0101" inside the tree of depth 4.

Then the code is uniquely decodable if no code word is father of another code word; in Fig. 1.2 it is easy to see that the node corresponding to code word “1100” is a son of node “11”, and therefore the 4 words given in Fig. 1.2 do not form a uniquely decodable code. This means that all the code words of a uniquely decodable code must be leaves of the tree.

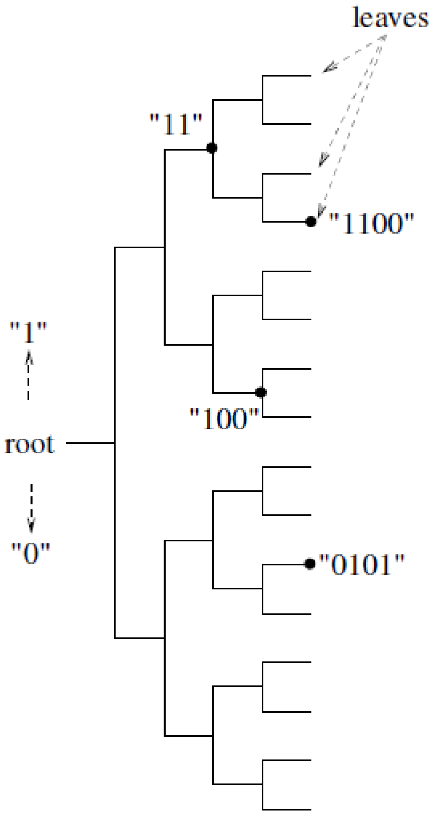


Figure 1.2: Example of association between code word “11”, “1100”, “100” and “0101” and nodes inside a binary tree.

THEOREM (Kraft inequality):

A uniquely decodable code exists **if and only if** the lengths n_k of its M code words satisfy the condition

$$\sum_{k=1}^{M} 2^{-n_k} \leq 1$$

PROOF:

We must actually give two proofs, one to one that the condition is sufficient to obtain a uniquely decodable code and one to show that the condition is necessary (i.e. no uniquely decodable codes exist for which $\sum_{k=1}^{M} 2^{-n_k} > 1$). We will only show in the following the sufficient condition, which is obtained through condition. Assume, without loss of generally, that $n_1 \leq n_2 \leq \cdots \leq n_M$, so that n_M is the maximum length of the code word. Consider a complete binary tree with depth $N = n_M$, and start by placing the first code word x_1 at a node with depth n_1 (corresponding to

n_1 bits). This first code word must be a leaf of the final tree, and therefore eliminate 2^{N-n_1} leaves of the complete tree. Place the second code word x_2 at a node with depth n_2, which removes 2^{N-n_2} leaves of complete tree, etc. The procedure stops if there is no node for code word x_j at depth n_j for some j ≤M. The number of available leaves after placing code word x_{j-1} is equal to the total number of leaves of the complete tree, which are 2^N , minus the leaves eliminated by code words $x_1, \dots, x_{j-1}$:

$$2^N - \sum_{k=1}^{j-1} 2^{(N-n_k)}$$

and it is not possible to place code word x_j if this number is smaller than $2^{(N-n_j)}$.

Thus the procedure stops only if, for some j ≤ M, the following inequality holds

$$2^N - \sum_{k=1}^{j-1} 2^{(N-n_k)} < 2^{(N-n_j)}$$

which corresponds to

$$1 - \sum_{k=1}^{j-1} 2^{-n_k} < 2^{-n_j}$$

or

$$\sum_{k=1}^{j} 2^{-n_k} > 1$$

But this is never the case because, by hypothesis,

$$\sum_{k=1}^{M} 2^{-n_k} = \sum_{k=1}^{j} 2^{-n_k} + \sum_{k=j+1}^{M} 2^{-n_k} = \sum_{k=1}^{j} 2^{-n_k} + A \leq 1$$

where A is an integer number. Then, for any j ≤ M

$$\sum_{k=1}^{j} 2^{-n_k} < 1 - A \leq 1$$

and it is never possible that

$$\sum_{k=1}^{j} 2^{-n_k} > 1$$

Then the procedure can be completed if (sufficient condition) $\sum_{k=1}^{j} 2^{-n_k} > 1$. The proof that this condition is also necessary is similar.

Let us now go back to the source coding theorem. In order to complete the proof, we must prove that, by choosing $\mathrm{I}(x = x_k) \leq n_k < \mathrm{I}(x = x_k) + 1$ we satisfy the Kraft inequality (we use the sufficient part of the Kraft inequality to prove that the uniquely decodable code exists). We have the following sequence of inequalities:

- $\mathrm{I}(x = x_k) \le n_k < \mathrm{I}(x = x_k) + 1$ Our choice of lengths
- $-\log_2 \pi_k \le n_k < -\log_2 \pi_k + 1$ We substitute the value of $\mathrm{I}(x = x_k)$
- $\log_2 \pi_k \ge -n_k > \log_2 \pi_k - 1 = \log_2 \frac{\pi_k}{2}$ Change sign
- $\pi_k \ge 2^{-n_k} > \frac{\pi_k}{2}$ Take 2 to the power of each of terms (the inequality signs are not changed since 2^x is a monotonic increasing function of x)
- $\sum_{k=1}^{M} \pi_k \ge \sum_{k=1}^{M} 2^{-n_k} > \sum_{k=1}^{M} \frac{\pi_k}{2}$ Sum all the terms (the inequality signs are not changed)
- $1 \ge \sum_{k=1}^{M} 2^{-n_k} > \frac{1}{2}$ (The sum of all the probabilities of the source symbol must be equal to 1, since the source certainly generates a symbol).

Then on the left of the above inequality, $1 \ge \sum_{k=1}^{M} 2^{-n_k}$, we exactly have the Kraft inequality, which means that we can build the uniquely decodable code starting from the chosen length value n_k (and the Kraft inequality tells us how to build it).

Then we proved that it is always possible, for any discrete source, to design a source encoder with average code word length $\bar{n}$ between $H(X)$, the source entropy, and $H(X) + 1$.

1.1.2.2.3 Source coding theorem (Shannon 1948)

In information theory, the source coding theorem (Shannon 1948) informally states that (MacKay 2003, pg. 81, Cover: Chapter 5):

> *N i.i.d. random variables each with entropy H(X) can be compressed into more than N H(X) bits with negligible risk of information loss, as N → ∞; but conversely, if they are compressed into fewer than N H(X) bits it is virtually certain that information will be lost.*

For a given alphabet $X = \{x_1, x_2, \dots, x_M\}$ and associated probabilities $\{\pi_1, \pi_2, \dots, \pi_M\}$, it is always possible, for any discrete source, to design a source encoder to find t unique decodable source code such that

$$\boldsymbol{H(X) \le \bar{n} \le H(X) + 1}$$

Proof:

- $\mathrm{I}(x = x_k) \leq n_k < \mathrm{I}(x = x_k) + 1$ For symbol x_k, pick a code word with length n_k
- $\mathrm{I}(x = x_k)\pi_k \leq n_k \pi_k < \pi_k \mathrm{I}(x = x_k) + \pi_k$ Multiple π_k on both sides. The sign does not change, because of nonnegative probability π_k.
- $\sum_{k=1}^{M} \mathrm{I}(x = x_k)\pi_k \leq \sum_{k=1}^{M} n_k \pi_k < \sum_{k=1}^{M} \pi_k \mathrm{I}(x = x_k) + \sum_{k=1}^{M} \pi_k$ Insert summation on both sides
- $H(X) \leq \bar{n} \leq H(X) + 1$ Source coding theorem is verified

Then, in the average, each bit at the output of an encoder built according to the source coding theorem carries an information quantity η

$$\eta = \frac{H(X)}{\bar{n}}$$

with

$$1 \geq \eta > \frac{H(X)}{H(X) + 1} = \frac{1}{1 + \frac{1}{H(X)}}$$

Noted that η is also called **efficiency of source code.** This means that, in the best case, each bit at the output of the encoder carries exactly one information bit and we can say that the source followed by the source encoder corresponds to a binary memoryless source with equally likely symbols “0” and “1”. In the worst case, each bit at the input of the encoder carries an information quantity larger than $\frac{H(X)}{H(X)+1}$; for H(X) sufficiently large, this bound can be approximated to 1, so that we can say that the system is equivalent to a binary source which almost memoryless and with equally likely symbols “0” and “1”, From the proof of source coding theorem, we see that $\eta = 1$ only if $n_k = -\log_2 \pi_k$, which is possible only if π_k is an integer negative power of 2.

It is interesting to analysis also the quantity

$$\nu = \frac{1}{\eta} = \frac{\bar{n}}{H(X)}$$

which represents the average number of bits (at the output of the source encoder) necessary to carry one information bit. Using again the same inequality, we get

$$1 + \frac{1}{H(X)} = \frac{H(X) + 1}{H(X)} > \nu \geq 1$$

which means again that, in the best case it is sufficient one bit to carry one information bit, but in the worst case we need $1 + 1/H(X)$ bits, which might be practically equal to 1, if $H(X)$ is sufficient large.

One way to increase the mean information quantity η of the bits at the output of the source encoder is to use a source encoder which does not encode one symbol at a time, but N consecutive symbols generated by the source. In fact, assuming the source is memoryless, then, N consecutive symbols have the entropy equal to $NH(X)$, being $H(X)$ the entropy of each symbol. According to the source coding theorem, we can design an encoder which satisfies this new inequality:

$$NH(X) \leq \overline{n_N} < NH(X) + 1$$

being $\overline{n_N}$ the average number of bits of a code word corresponding to N input symbol. Then it is as if each source symbol is encoded with

$$\bar{n} = \frac{\overline{n_N}}{N} \text{ bits}$$

with

$$\boldsymbol{H(X) \leq \bar{n} < H(X) + \frac{1}{N}}$$

which shows that the upper bound for $\bar{n}$ can be arbitrarily decreased to $H(X)$ by taking N sufficiently large.

This last is the justification of the assuming which typically made when digital modulators are studied, i.e. that the modulator input bits are statistically independent and with equally probabilities. This is true if we assume that a sufficiently strong source encoder is used, such that each bit carries practically one information bit.

1.1.2.3 Huffman coding and variable length source codes

The source coding theorem gives us a theoretical way of designing an efficient source encoder, but the design might be impractical in many cases, due to a large value of n_M (the longest code word) or a large number of symbols.

Much more practical is the encoding technique described by Huffman (Huffman encoder), which was proved to provide the optimum code word mapping, i.e. the one with minimum possible value of $\bar{n}$ for the given source.

The source encoder for a generic source with symbols $\{x_1, x_2, \ldots, x_M\}$ having probabilities $\pi_1, \pi_2, \ldots, \pi_M$, can be build according to this procedure (see Fig. 1.3):

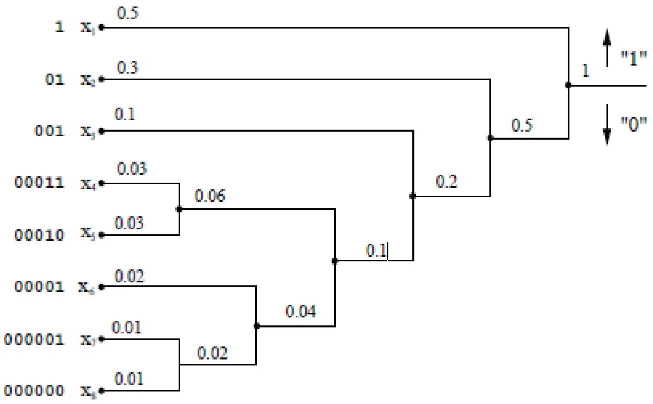


Figure 1.3: Example of Huffman code tree for the symbols and probabilities listed in table 1.2

Step 1: Order symbols so that the symbol with higher probability is on the top ("bubble sorting"); without loss of generality, let x_1 be the most likely symbol and x_M the less likely symbol.
Step 2: Group the two symbols with the smallest probabilities (the two on the bottom) by drawing two branches which merge in a node; this node corresponds to an equivalent symbol with probability $\pi_M + \pi_{M-1}$. The equivalent symbol corresponds to the event "the source generates π_M or π_{M-1}".
Step 3: Decrease M by 1.
Step 4: Repeat steps 1-3 until you have just one equivalent symbol with probability 1.
Step 5: At this point you have a tree in which the symbols are the leaves; associate a code word to each symbol as follows: start from the root and follow the path which leads to the desired leaf/symbol, each branch going up corresponds to bit "1", each branch going down corresponds to bit "0" (the first bit of the code word is the one related to the branch that merges in the root of the tree).

The source encoder uses a look-up table which gives the correspondence between symbols and code words. The decoder can use the tree to perform decoding. It starts from the root and follows the path specified by the input bits: an input bit equal to "1" means "take the upper branch", an input bit equal to "0" means "take the lower branch". Once the decoder reaches the leaf of the tree, it generates the corresponding symbol, and starts again from the root. Note that the Huffman code is uniquely decodable by construction: no code word is prefix of another code word.

An example of application of the Huffman encoding is shown in figure 1.3; the obtained code words are listed in table 1.2

Symbol x_i	Probability π_i	$I(x_i)$	Code word	Code word length n_i
x_1	0.50	1.0	1	1
x_2	0.30	1.7369	01	2
x_3	0.10	3.3219	001	3
x_4	0.03	5.0589	00011	5
x_5	0.03	5.0589	00010	5
x_6	0.02	5.6438	00001	5
x_7	0.01	6.6438	000001	6
x_8	0.01	6.6438	000000	6

Table 1.2: Symbols and code words for the Huffman code shown in Fig.1.3

For this example, the source entropy is

$$H(x) = \sum_{k=1}^{8} \pi_k I(x = x_k) \ = 1.9025\, information\, bits$$

whereas the mean code word length is

$$\overline{n} = \sum_{k=1}^{8} \pi_k n_k \ = 1.92$$

The Huffman code allows getting the minimum value of $\overline{n}$, which is, however, larger than $H(X)$, since the probabilities are not all integer negative powers of 2. In this example, $\eta = 1.09025/1.92 = 0.991$ which is very close to 1; we can say that the Huffman encoder generates bits which are practically equally likely and statistically independent even if the source symbols have very different probabilities. Note also that the source code we can build using the technique described in the proof of the source coding theorem is less efficient than the Huffman code.

Therefore, the output from Huffman's algorithm can be viewed as a variable-length code table for encoding a source symbol. The algorithm derives this table from the estimated probability or frequency of occurrence (weight) for each possible value of the source symbol. As in other entropy encoding methods, more common symbols are generally represented using fewer bits than less common symbols.

1.1.3 Application of information theory to some games

Let us consider again the game described in Sect. 1.1.1.1 and let us try to solve it using the Huffman encoder. There are 4 symbols with probabilities $1/2, 1/4, 1/8, 1/8,$ and we have to guess which of them has been generated by the source, using, in the average, the minimum number of questions with answers “yes/no”. Then this is equivalent to finding the minimum average number of bits which encode the 4 symbols, and we can use the Huffman encoding technique, which provides the same solution described in section 1.1.1.1. Let us consider instead the case in which there are the 8 symbols with the 8 probabilities listed in table 1.2: we apply the Huffman code and we get the tree of Fig. 1.3: How should we interpret that tree in terms of questions that B has to ask A? The questions are the following:

1. Is the symbol x_1? With probability 0.5 the answer is "yes" and the game stops, otherwise B moves to question 2.
2. Is the symbol x_2? With probability $0.3/0.5$ the answer is "yes" and the game stops, otherwise B moves to question
3. Is the symbol x_3? With probability $0.1/0.2 = 0.5$ the answer is "yes" and the game stops, otherwise B moves to question 4.
4. Is the symbol x_4 or x_5? With probability $0.06/0.1 = 0.6$ the answer is "yes" and B asks the subsequent question 4.1: is the symbol x_4? Which ends the game whatever is the answer? If the answer to question 4 is "no", then B goes to question 5.
5. Is the symbol x_6? With probability $0.02/0.04 = 0.5$ the answer is "yes" and the game stops, otherwise B asks question 6.
6. Is the symbol x_7? Whatever is the answer, the game stops.

Consider the game in which there are N coins, out of which is false and you want to find it out. Sometimes you know if the false coin weighs more or less than the true coins, sometimes you have to find it out. Sometimes you have a person to whom you have to ask questions with answers "yes/no", sometimes you have a scale and you have to decide how to use it. There are many variants of the game, but in many cases you can find the solution of the problem using information theory.

For example, consider the case in which you know that a correct coin weighs 10 g, while the false coin has a weight equal to 9 g. You have 10 coins, you don't know if false coins are among them. Assume that you have a normal digital spring scale with infinite precision. Then, from a theoretical point of view, the quantity of information that the scale gives you each time you make a measurement is infinite, and, whatever is the problem, a solution should exist so that it is sufficient to use the scale just once to find the answer. In the current case, you can simply put all the coins on the scale:

- if all the coins are true, the weight is 100 g
- if one (and only one) coin is false, then the weight is 99 g
- if exactly two coins are false, then the weight is 98 g
- etc.

Then, if w is the measured weight, the number of false coins will be $100 - w$. However, you do not know which coins are false.

Let us consider a similar problem, in which you have the same scale, 10 coins each with nominal weight 10 g, you know that only 1 coin can be false, but you don't know whether it weighs more or less than a true coin. Then you can again put all the coins on the scale and get the total weight w. If $w = 100$g, all the coins are true; if $w = (100 + |\in|)$, then one false coin is present and it weighs $(10 + |\in|)$g; if $w = (100 - |\in|)$ (it weighs more), then one false coin is present and it weighs $(10 - |\in|)$g (it weighs less).

Let us consider the problem in which you have 10 coins, out of which K are false, but you don't know the value of K and the weight of the false coin. In this case, it is convenient to use concepts of linear algebra: you have two unknowns K and $\in$, and it is not possible to find their values using just one equation. If the total weight is $w = 102$g, how can we know if there are 2 false coins and $\in = 1$g, or there are 4 false coins and $\in = 0.5$g? The point is that, in principle, $\in$ can take any value, so that its entropy depends on its probability density function, and the infinite information quantity provided by the scale is not sufficient to provide also the information required to find out K (actually K can take only 11 values, from 0 to 10, and its entropy is $\log_2 11$, since all the possibilities are equally likely). In real life, however, the value of $\in$ will be small, not infinitesimal nor infinite, and a solution might be found but we need to know something more about $\in$.

Let us consider the case in which you have M coins, one is false, your friend knows which one is false and you have to ask him questions with answers "yes/no" to find the false coins with the (mean) minimum number of questions. Then the information quantity you must obtain from your friend is $1 + \log_2 M$ (where $\log_2 M$ is necessary to find the false coin, and 1 bit is necessary to find whether it weighs more or less than a true coin), your friend answer carries at most one information bit, the minimum (theoretical) number of questions is $1 + \log_2 M$, and the Huffman code can be directly applied to find the optimum solution.

Let us consider the case in which you have M coins, one is false, you can use only a balance scale. Then the information quantity you need is still $1 + \log_2 M$, and the balance scale gives you at most $\log_2 3$ information bits (the scale has three possible answers: the weight on the right is larger than the weight on the left, the weight on the left is larger than the weight on the right, and two weights are equal). From a theoretical point of view the (mean) minimum number of measurements is $1 + \log_2 M = \log_2 3$. In this case you need a ternary Huffman code (three branches from each node), and the interpretation of the tree is much more complex. In any case you should find a way to get from the balance the maximum quantity of information each time you use it; the maximum information quantity is $\log_2 3$ bits, which means that you must devise an experiment/measurement with three equally likely outcomes.

1.2 Mutual information

1.2.1 Discrete channel matrix

In the previous section we discussed source encoding, using information theory, and we assumed that the subsystem between the source encoder and decoder is ideal. Now we want to consider this inner subsystem, made of the digital modulator, the channel and the demodulator.

Basically, the modulator maps the input symbol into a waveform, suitable for the specific channel (at baseband, at radio-frequency, with a given bandwidth, etc). The waveform travels through the channel which typically adds noise and sometimes introduces distortions. The demodulator performs the opposite operation of the modulator and maps the received waveform into one of the output symbols. For example, a 4PSK modulator has input x which takes one out of 4 possible values, which we can simply identify as x_1, x_2, x_3, x_4. The standard 4PSK demodulator outputs a symbol y which takes one out of 4 values y_1, y_2, y_3, y_4, and $y_1 = x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4$; of course we would like that, being for example x_2 the symbol at the input of the modulator, the output of the demodulator is y_2, which means that the symbol was correctly receiver. However, we can imagine a non-standard 4PSK demodulator which outputs a symbol y taken from a set of 8 possible symbols, and it is thus convenient to use in general two different alphabets for the inputs of the modulator and for the outputs of the demodulator.

Then, in general the modulator has an input x from the set $X = \{x_1, x_2, \dots, x_{N_T}\}$, whereas the demodulator outputs a symbol y taken from the set $Y = \{y_1, y_2, \dots, y_{N_R}\}$, and it is possible that $N_R \neq N_T$. Whatever is the channel, there is a probability that an input symbol x_k is received as y_n for any $k \in [1, N_T]$ and any $n \in [1, N_R]$. The conditional probability

$$P(y = y_n | x = x_k) \geq 0$$

can be zero, but in general it is positive. Of course,

$$\sum_{n=1}^{N_R} P(y = y_n | x = x_k) = 1$$

since an input symbol x_k must correspond to an output symbol.

The conditional probabilities $P(y = y_n | x = x_k)$ are conveniently arranged into a matrix, which is called the **discrete matrix $\boldsymbol{P}$**, where the adjective "discrete" is used to emphasize the fact that it is related to digital modulations with a finite number of symbols:

$$
\boldsymbol{P} = \begin{bmatrix} P(y_1|x_1) & P(y_2|x_1) & P(y_3|x_1) & & P(y_{N_R}|x_1) \\ P(y_1|x_2) & P(y_2|x_2) & P(y_3|x_2) & \cdots & P(y_{N_R}|x_2) \\ P(y_1|x_3) & P(y_2|x_3) & P(y_3|x_3) & & P(y_{N_R}|x_3) \\ & \vdots & & \ddots & \vdots \\ P(y_1|x_{N_T}) & P(y_2|x_{N_T}) & P(y_3|x_{N_T}) & \cdots & P(y_{N_R}|x_{N_T}) \end{bmatrix}
$$

The channel matrix is rectangular, with N_T rows and N_R columns.

1.2.1.1 Symmetric channels

Looking at the channel matrix ***P***, it is possible to decide if the channel is symmetric or not, by applying the following definition. A channel is symmetric if its channel matrix has the following properties:

1. either the following two conditions are satisfied (both must be satisfied and we have a **strict-sense symmetric channel**)
   - Each row of ***P*** has the same set of values (all the numbers in the first row are present in the second row, in the third, etc.)
   - Each column of ***P*** has the same set of values (all the numbers in the first column are present in the second column, in the third, etc.)
2. Or (**wide-sense symmetric channel**) matrix ***P*** (with dimension N_T times N_R) can be divided into two or more submatrices of dimension $N_T \times N_R'$ ($N_R' < N_R$) such that each of the submatrices satisfy the properties in item 1. Note that it is always possible to exchange two columns of ***P*** (which means that the output symbols are simply re-ordered), or the rows of ***P*** (which means that the input symbols are re-ordered).

1.2.2 The entropies associated with the discrete channels

The discrete channels are characterized by an input alphabet $X = \{x_1, x_2, \dots, x_{N_T}\}$ and an output alphabet $Y = \{y_1, y_2, \dots, y_{N_R}\}$, corresponding to the discrete channel matrix and it is possible to define the entropies for these two alphabets. With the following definitions:

$$
\pi_i = P(x = x_i) \qquad \rho_j = P(y = y_j)
$$

We can define the following entropies:

- *Input entropy*

$$
H(X) = E\{I(x)\} = -\sum_{i=1}^{N_T} \pi_i \log_2 \pi_i
$$

The input entropy is the mean quantity of information that the source produces each time it generates a new symbol, but this is also the mean quantity of information that the receiver needs, to exactly know which symbol x was the source output/channel input.

- *Output entropy*

$$H(Y) = E\{I(y)\} = -\sum_{j=1}^{N_R} \rho_j \log_2 \rho_j$$

The output entropy is the mean quantity of information of each received symbol.

- *Joint entropy*

$$H(X,Y) = E\{I(x,y)\} = -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \log_2 \mathrm{P}(x_i, y_j)$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)P(x_i) \log_2[\mathrm{P}(y_j|x_i)P(x_i)]$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)\pi_i \log_2[\mathrm{P}(y_j|x_i)\pi_i]$$

The joint entropy is the mean quantity of information of the couple (x, y).

- *Conditional entropy with condition on x*

$$H(Y|X) = E\{I(y|x)\} = -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j, x_i) \log_2 \mathrm{P}(y_j|x_i)$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)P(x_i) \log_2[\mathrm{P}(y_j|x_i)]$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)\pi_i \log_2[\mathrm{P}(y_j|x_i)]$$

The conditional entropy is the mean quantity of information carried by the output symbol y, being known the input symbol x. If, starting from an input symbol x_k, it is possible to obtain more than one output symbol, then $H(Y|X) > 0$ (the channel introduces uncertainly); if, on the contrary, for each input symbol x_i, there is only one output symbol y_j, then $H(Y|X) = 0$, since it is possible to exactly predict y knowing x.

- Conditional entropy with condition on y or **equivocation**

$$H(X|Y) = E\{I(x|y)\} = -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)\log_2 \mathrm{P}(x_i|y_j)$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)P(x_i)\log_2[\mathrm{P}(x_i|y_j)]$$
$$= -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(y_j|x_i)\pi_i\log_2[\mathrm{P}(x_i|y_j)]$$

This conditional entropy is the mean quantity of information carried by the input symbol x, being known the output symbol y. In this case, the situation is that of the typical receiver, which knows y and has to guess which was x. It is better to interpret $H(X|Y)$ as the uncertainty that the receiver has on which x was transmitted, once the receiver has observed y; on the other hand, $H(X)$ can be interpreted as the uncertainty that the receiver has on x, without having observed the channel output y.

An **ideal channel** must have $H(X|Y) = 0$. Once the channel output has been observed, the receiver has no uncertainty on the transmitted symbol. A **"normal" channel** has $H(X|Y) < H(X)$), since the observation of the channel output carries some information on the transmitted symbol and therefore the receiver uncertainty is reduced. A **"useless" channel** has $H(X|Y) = H(X)$: the receiver uncertainty is the same, whether it observes the channel output, and therefore the symbol transmission is a waste of energy and resources, the channel is useless.

We can think of $H(X|Y)$ as the mean quantity of information that the receiver must get (somehow) in order to remove all its uncertainty and know exactly which symbol was transmitted. If a demon (like the Maxwell's demon) were present, then the receiver could ask the demon some questions (with answers yes/no) to know the value of x, and in the average, with an ideal game, the number of questions would be $H(X|Y)$.

Note that

$$P(x_i|y_j) = \frac{P(x_i, y_j)}{P(y_j)} = \frac{P(y_j|x_i)P(x_i)}{P(y_j)} = P(y_j|x_i)\frac{\pi_i}{\rho_j}$$

and that

$$\rho_j = \sum_{i=1}^{N_T} P(y_j|x_i)\pi_i$$

Therefore, the above entropies depend on the channel matrix **P** whose i, j element is $P(y = y_j|x = x_i)$ and on the probabilities π_i. Some relationships exist among the entropies, since they depend on the same quantities.

- A first relationship is easily found using the Bayes' rule:

$$\mathrm{P}(x = x_i, \mathrm{y} = y_j) = \mathrm{P}(\mathrm{y} = y_j | x = x_i)\mathrm{P}(x = x_i) = \mathrm{P}(x = x_i | \mathrm{y} = y_j)\mathrm{P}(y = y_j)$$

from which, it is possible to obtain the following relationship in terms of information quantity:

$$\mathrm{I}(x = x_i, \mathrm{y} = y_j) = \mathrm{I}(\mathrm{y} = y_j | x = x_i) + \mathrm{I}(x = x_i)$$

$$= \mathrm{I}(\mathrm{x} = x_i | y = y_j) + \mathrm{I}(y = y_j)$$

Taking the mean of the above equation over all the possible values of x_i and y_j, we get

$$\begin{aligned} \mathrm{H}(X, Y) = E\{I(x, y)\} &= E\{I(y|x)\} + E\{I(x)\} = H(Y|X) + H(X) \\ &= E\{I(x|y)\} + E\{I(y)\} = H(X|Y) + H(Y) \end{aligned}$$

Note that the evolution of $H(Y|X)$ is in general easier than the evolution of $H(X|Y)$, and the fact that $H(Y|X) + H(X) = H(X|Y) + H(Y)$ can be used to reduce the computation burden for $H(X|Y)$.

- A second couple of relationships is the following (already partially described in words)

$$H(X|Y) \le H(X) \qquad H(Y|X) \le H(Y)$$

The formal proof of the inequality $H(X|Y) \le H(X)$:
Since

$$P(x_i) = \sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)$$

We can write

$$H(X|Y) - H(X) = \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \log_2 \frac{1}{P(x_i|y_j)} - \sum_{i=1}^{N_T} P(x_i) \log_2 \frac{1}{P(x_i)}$$

$$= \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \log_2 \frac{1}{P(x_i|y_j)} - \sum_{i=1}^{N_T} \left[\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)\right] \log_2 \frac{1}{P(x_i)}$$

$$= \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \log_2 \frac{P(x_i)}{P(x_i|y_j)} = \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \frac{\ln\left[\frac{P(x_i)}{P(x_i|y_j)}\right]}{\ln 2}$$

$$= \frac{1}{\ln 2}\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \ln\left[\frac{P(x_i)}{P(x_i|y_j)}\right]$$

$$\le \frac{1}{\ln 2}\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \left[\frac{P(x_i)}{P(x_i|y_j)} - 1\right]$$

$$= \frac{1}{\ln 2}\left[\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \frac{\mathrm{P}(x_i, y_j)}{P(x_i|y_j)} P(x_i) - \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)\right]$$

$$= \frac{1}{\ln 2}\left[\sum_{i=1}^{N_T} P(x_i) \sum_{j=1}^{N_R} \frac{\mathrm{P}(x_i, y_j)}{P(x_i|y_j)} - \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)\right]$$

Since, based on the Bayes' rules:

$$P(y_j) = \frac{P(x_i|y_j)P(y_j)}{P(x_i|y_j)} = \frac{P(x_i, y_j)}{P(x_i|y_j)}$$

And

$$\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) = \sum_{i=1}^{N_T} P(x_i) = \sum_{j=1}^{N_R} P(y_j) = 1$$

Then we can write

$$H(X|Y) - H(X) \le \frac{1}{\ln 2}\left[\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} P(y_j)\, P(x_i) - 1\right]$$

$$= \frac{1}{\ln 2}\left[\sum_{i=1}^{N_T} P(x_i) \sum_{j=1}^{N_R} P(y_j) - 1\right] = \frac{1}{\ln 2}[1 - 1] = 0$$

Obviously, we proved

$$H(X|Y) - H(X) \le 0$$

i.e.

$$H(X|Y) \le H(X)$$

1.2.3 Mutual information

Mutual information is the mean quantity of information that the channel carries from the input source X to the output of the receiver Y. The average information quantity that the receiver needs (from the Maxwell demon) to exactly know the transmitted symbol is $H(X)$ in the absence of channel, and $H(X|Y) < H(X)$ in the presence of channel: this means that the channel provides the receiver with $H(X) - H(X|Y)$ information bits and this is the average flow of information through the channel, which we call the **mutual information** $I(X;Y)$:

$$I(X;Y) = H(X) - H(X|Y) \ \ [information\ bits]$$

In the other way, based on the expression of conditional entropy (section 1.2.2), the mutual information between two variables can be defined as:

$$I(X;Y) = -\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \log_2 \frac{\mathrm{P}(x_i|y_j)}{\mathrm{P}(x_i)\mathrm{P}(y_j)}$$

More generally,

$$I(X_1, X_2, \dots, X_n; Y_1, Y_2, \dots, Y_m) = I\big(X_{\pi(1)}, X_{\pi(2)}, \dots, X_{\pi(n)}; Y_{\pi'(1)}, Y_{\pi'(2)}, \dots, Y_{\pi'(m)}\big)$$
$$= I(Y_{\pi'(1)}, Y_{\pi'(2)}, \dots, Y_{\pi'(m)}; X_{\pi(1)}, X_{\pi(2)}, \dots, X_{\pi(n)})$$

for any pair of permutations π, π'.

Since $H(X|Y) \le H(X)$, we get $I(X;Y) \ge 0$. I.e. the mutual information is nonnegative number. The proof is following:

$$-I(X;Y) = \log_2 e \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \ln \frac{\mathrm{P}(x_i)\mathrm{P}(y_j)}{\mathrm{P}(x_i|y_j)}$$
$$\le \log_2 e \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j) \left(\frac{\mathrm{P}(x_i)\mathrm{P}(y_j)}{\mathrm{P}(x_i|y_j)} - 1\right)$$
$$= \log_2 e \left(\sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i)\mathrm{P}(y_j) - \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \mathrm{P}(x_i, y_j)\right) = 0$$

Notice that the mutual information is invariant to the exchange of random variables. For example, $I(X;Y) = I(Y;X)$. The proof is following:

$$I(X;Y) = H(X) - H(X|Y) = H(X) - [H(X|Y) + H(Y)] + H(Y)$$
$$= H(X) + H(Y) - H(X,Y) = H(X) + H(Y) - [H(Y|X) + H(X)]$$
$$= H(Y) - H(Y|X) = I(Y;X)$$

This proof verifies that the information flow is bidirectional: it can be evaluated in the direction from X to Y or in the direction from Y to X. In general, it is convenient to use the formula

$$I(X;Y) = H(Y) - H(Y|X)$$

We can see that:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Then, we define the mutual information between continuous random variables by simply replacing the discrete entropies by corresponding differential entropies:

$$I(X;Y) = h(X) + h(Y) - h(X,Y)$$

Moreover, we can use an analogy taken from hydraulics, where pipes transport water, whereas telecommunication channels transport information. The direct channel behaves like a pipe and the overall transported information depends on how much information $H(X)$ is present at the input of the channel and how much information $H(X|Y)$ is lost in the channel. If a pipe has an input water flow, then the output flow is equal to the input flow minus the water flow that is lost due to understand but always present leaks.

For the **ideal channel**, we have $H(X|Y) = 0$ and thus the mutual information $I(X;Y) = H(X)$: the ideal channel carries all the information generated by the source.

For the **useless channel**, we have $H(Y) = H(Y|X)$ and the mutual information is $I(X;Y) = 0$, which means that it cannot carry any information at all.

1.2.3.1 The symmetric channel

In the case of symmetric channel, the evaluation of $I(X;Y)$ is following:

$$H(Y|X) = -\sum_{i=1}^{N_T} P(x_i) \sum_{j=1}^{N_R} P(y_j|x_i) \log_2 P(y_j|x_i)$$

The summation over j depends on the elements of the i^{th} row of the channel matrix ***P***; but for a symmetric channel each row has the same elements (maybe with a different order) and therefore the sum does not depend on i. Then we can write:

$$H(Y|X) = -\left[\sum_{i=1}^{N_T} P(x_i)\right] \sum_{j=1}^{N_R} P(y_j|x_1) \log_2 P(y_j|x_1) = \sum_{j=1}^{N_R} P(y_j|x_1) \log_2 P(y_j|x_1) = A$$

which depends only on the channel, not on the a-priori probabilities $P(x_i) = \pi_i$. In order to evaluate $I(X;Y)$, we still need $H(Y)$, which depends on the probabilities $P(y_j) = \rho_j$:

$$\rho_j = \sum_{j=1}^{N_R} P(y_j|x_i) \pi_i$$

Then ρ_j is evaluated using the j^{th} column of ***P***: for a symmetric channel the elements of each column are equal (at least in each submatrix), but this property does not help since ρ_j also depends on π_i. In any case the mutual information for a symmetric channel can be written as

$$I(X;Y) = H(Y) - \sum_{j=1}^{N_R} P\left(y_j \middle| x_1\right) \log_2 P\left(y_j \middle| x_1\right)$$

and it depends on the channel and on the a-priori probability π_i.

1.3 Capacity

1.3.1 Capacity of a discrete channel

We have seen the mutual information $I(X;Y)$ depends on the channel itself and on the probabilities $\pi_i = P(x = x_i)$. This is natural if we think of the definition of mutual information:

$$I(X;Y) = H(X) - H(X|Y)$$

in which $H(X)$ (depends on π_i) appears explicitly.

We can use again the hydraulics analogy. If a pipe has a small input flow, then the output flow is also small, and if we want to increase the output flow we must increase the input flow. Actually not all the water at the input of the pipe gets to the output, since some water is lost due to undesired but always present leaks. Each water pipe has a capacity (liters per second) and if the input flow is larger than this capacity, typically the pipe breaks. An experimental way to measure the capacity of a pipe is that of increasing the input flow until the output flow reaches a maximum (hopefully without breaking the pipe). Then, in analogy, we can define the capacity of a discrete channel as the maximum mutual information (or information flow) obtained by varying the input entropy, i.e. by varying the a-priori probabilities $\pi_1, \pi_2, \ldots, \pi_{N_T}$:

$$\mathrm{C} = \max_{\{\pi_1, \pi_2, \ldots, \pi_{N_T}\}} I(X;Y)$$

We can guess that, at least in some cases, the capacity is obtained when $H(X)$ is maximum, i.e. when $\pi_i = 1/N_T$. The example is showed in section 1.3.1.1.

1.3.1.1 The symmetric channel

We have seen that the mutual information of a symmetric channel can be written as

$$I(X;Y) = H(Y) - \sum_{j=1}^{N_R} P\left(y_j \middle| x_1\right) \log_2 P\left(y_j \middle| x_1\right)$$

and capacity is

$$C = \max_{\pi} I(X;Y) = \left[\max_{\pi} H(Y)\right] - \sum_{j=1}^{N_R} P(y_j|x_1)\log_2 P(y_j|x_1)$$

Since the second term does not depend on π_k, but only on the channel matrix $\boldsymbol{P}$. Then, for the specific case of symmetric channels, the capacity is obtained by maximizing the output entropy $H(Y)$.

If $\boldsymbol{P}$ has the same elements in all its columns (condition 1 in the definition of symmetric channels, case of strict-sense symmetric channels), then the maximum of $H(Y)$ is obtained by setting $\pi_i = 1/N_T$. In fact the maximum value of $H(Y)$ is $\log_2 N_R$, and you can get it if and only if $\rho_j = 1/N_R$ for all $j = 1, \dots N_R$; but if $\pi_i = 1/N_T$, then we have

$$\rho_j = \sum_{i=1}^{N_T} P(y_j|x_i)\,\frac{1}{N_T} = \left(\sum_{i=1}^{N_T} P(y_j|x_i)\right)\frac{1}{N_T}$$

and this result does not depend on j, since all the column of $\boldsymbol{P}$ have the same set of probabilities $P(y_j|x_i)$. Then we can write:

$\rho_j = \left(\sum_{i=1}^{N_T} P(y_j|x_i)\right)\frac{1}{N_T} = A$ Constant, independent on j

Since $\rho_j = A$ for $j = 1, \dots N_R$ and it must be that $\sum_{i=1}^{N_R} \rho_j = 1$, then the only solution is $\rho_j = 1/N_R$. Then, if the channel is symmetric and $\pi_i = 1/N_T$ for all $j's$, we will have $H(X) = \log_2 N_T$ and $H(Y) = \log_2 N_R$ (maximum entropies at both the input and output of the channel). Note also that, if the channel matrix is doubly stochastic, then $\sum_{i=1}^{N_T} P(y_j|x_i) = 1$ for all j, and therefore $\rho_j = 1/N_T$.

In a summary, the capacity of strict-wise symmetric channel is

$$C = \log_2 N_R - \sum_{j=1}^{N_R} P(y_j|x_1)\log_2 P(y_j|x_1)$$

And it is obtained for $\pi_k = 1/N_T, k = 1, \dots, N_T$.

If the channel matrix is that of a wide-sense symmetric channel (condition 2 in the definition of symmetry), then it can be shown that the capacity is obtained for $\pi_k = 1/N_T, k = 1, \dots, N_T$ (i.e. $H(X) = \log_2 N_T$), but in this case, the output symbol are no more equally likely (i.e. $\rho_j = 1/N_R$) and the entropy $H(Y)$ has to be evaluated (it will be $H(Y) < \log_2 N_R$).

In the following we evaluate the capacity of some discrete channels, and, in the last section, the capacity of the analog AWGN channel.

1.3.2 Capacity of an AWGN channel

Now we study the capacity of the analog AWGN channel, assuming that we do not use any digital modulator. Then the situation is the following (see figure 1.4):

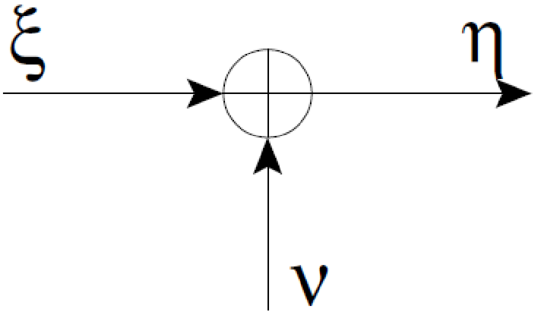


Fig.1.4: Block diagram for the AWGN channel

- An analog symbol ξ with a given probability density function $f_{\xi}(x)$ is transmitted over the channel; it is assumed that the variation of ξ is equal to $\sigma_{\xi}{}^{2}$ and the mean μ_{ξ} is zero, but there are no further restriction on $f_{\xi}(x)$.
- The AWGN channel adds ν, a Gaussian random variable with variance $\sigma_{\nu}{}^{2}$ and mean value zero (the probability density function of ν is denoted as $f_{\nu}(x)$).
- The receiver gets $\eta = \xi + \nu$, a random variable with probability density function $f_{\eta}(x) = f_{\xi}(x) * f_{\nu}(x)$ (where $*$ stands for convolution).

For the case of an analog AWGN channel, the capacity is obtained by maximizing just $h(\eta)$. But we know from section 1.1.1.2 that the maximum entropy of an analog source X is obtained when the analog source has a Gaussian probability density function. In particular we showed that, for a Gaussian source x

$$h(x) = \frac{1}{2}\log_2(2\pi e \sigma_x{}^2)$$

where $\sigma_x{}^2$ is the variance of x. In this case, if the source ξ is Gaussian with zero mean, then also $\eta = \xi + \nu$ is Gaussian, being the sum of two statistically independent Gaussian random variable, and η has mean equal to the sum of the means and variance equal to the sum of variance of ξ and ν. So if ξ has variance

$$h(\eta) = \frac{1}{2}\log_2[2\pi e(\sigma_{\xi}{}^2 + \sigma_{\nu}{}^2)]$$

and this is the maximum value of $h(\eta)$ (for the given and fixed $\sigma_{\xi}{}^2$).

Let us then complete the evaluation of the conditional entropy $h(\eta|\xi)$.

$$f_{\eta|\xi}(y|x) = \frac{1}{\sqrt{2\pi\sigma_{\nu}{}^2}} exp\left\{-\frac{(y-x)^2}{2\sigma_{\nu}{}^2}\right\}$$

$$h(\eta|\xi) = -\int_{-\infty}^{\infty} f_{\eta|\xi}(w|u)\log_2 f_{\eta|\xi}(w|u)dw = \left(\frac{1}{2}\log_2[2\pi e\sigma_{\nu}{}^2]\right)$$

In the overall, when the input is Gaussian mutual information is

$$I(\xi;\eta)=\frac{1}{2}\log_2\left[2\pi e\left({\sigma_\xi}^2+{\sigma_v}^2\right)\right]-\frac{1}{2}\log_2[2\pi e{\sigma_v}^2]=\frac{1}{2}\log_2\frac{{\sigma_\xi}^2+{\sigma_v}^2}{{\sigma_v}^2}$$

But this is also the c**apacity of the AWGN channel**:

$$C=\frac{1}{2}\log_2\frac{{\sigma_\xi}^2+{\sigma_v}^2}{{\sigma_v}^2}$$

So, each time the AWGN channel is used, it carries at most C information bits, and C depends on the signal to noise ratio $\frac{{\sigma_\xi}^2}{{\sigma_v}^2}$: if the noise variance reduces or the source increases, then the capacity increases.

Let us consider now not just the transmission of one analog symbol ξ, but a sequence of symbols, and let us limit the problem to the case of a bandlimited channel, in particular a low-pass channel with bandwidth B. Then, only a process $\xi(t)$ with bandwidth at most equal to *B* can pass through the channel without being distorted, and we can represent the information content of $\xi(t)$using just its samples, taken at sampling frequency *2B*[6]. Then the entropy of the Gaussian source is

$$h\big(\xi(t)\big)=2Bh(\xi)=B\log_2(2\pi e{\sigma_\xi}^2)$$

where ${\sigma_\xi}^2$ is the variance of the process; remember that, if the process is statistically and ergodic, which we will assume then ${\sigma_\xi}^2$ does not change with time and is equal to mean power P_ξ of the process.

The channel output process $\eta(t)$ is the sum of $\xi(t)$ and the white Gaussian noise $v(t)$ having power spectral density $N_0/2$. The receiver has an initial low pass filter followed by a sampler at frequency *2B*, so that we can write that the input of the detector is a sequence of samples, generated as rate *2B* samples per seconds, which are the sum of the samples of $\xi(t)$ and noise random variables with variance

$${\sigma_v}^2=\frac{N_0}{2}2B=N_0B$$

The entropy of $\eta(t)=\xi(t)+v(t)$, sampled at rate *2B*, is

$$h\big(\eta(t)\big)=2Bh(\eta)=B\log_2[2\pi e({\sigma_\xi}^2+{\sigma_v}^2)]$$

The conditional entropy is $h\big(\eta(t)\big|\xi(t)\big)=B\log_2[2\pi e{\sigma_v}^2]$ as before.
The AWGN channel capacity is then

$$C'=h\big(\eta(t)\big)-h\big(\eta(t)\big|\xi(t)\big)=B\log_2\frac{{\sigma_\xi}^2+{\sigma_v}^2}{{\sigma_v}^2}$$

We can substitute the values of the variances and obtain

[6] According to the sampling theorem that states that if a signal $x(t)$ has bandwidth *B*, it is possible to exactly evaluate $x(t)$ from its samples, provided that the sampling frequency is larger than $2B_x$.

$$C' = B\log_2\left(1+\frac{P_\xi}{N_0B}\right)$$

where now the unit of measure of $C'$is bits of information per second (not just bit of information).

In brief, compare to C and C':

- C is capacity per **channel use**

  $C = \frac{1}{2}\log_2\left(1+\frac{P_\xi}{N_0B}\right)$ *[information bit per channel use]*

- C' is capacity **measured**, which use the channel *2B* times per second. If we do not use the low pass filter, the capacity is zero for sure.

  $C' = B\log_2\left(1+\frac{P_\xi}{N_0B}\right)$ *[information bit per second]*

Let us see if we can relate the discrete channel capacities with the AWGN channel capacity. We can imagine that process $\xi(t)$ is the output of a digital modulator that generates bits (real bits "1" or "0") at rate R_b bits/s, so that the power P_ξ can be expressed as $P_\xi = \frac{E_b}{T_b} = E_bR_b$, where E_b is the energy per bit. So we have, for the AWGN channel, $C' = B\log_2\left(\frac{E_bR_b}{N_0B}+1\right)$ or $\frac{C'}{B} = \log_2\left(\frac{E_bR_b}{N_0B}+1\right)$.

It is not possible to get an error probability equal to zero if the input entropy is larger than the channel capacity. At most one bit transmitted by the digital modulator carries one information bit, so that we can say that the source entropy is $H(X) = R_b$ information bits per second, and, if we assume that we are **working at the limit**, i.e. the best case, with $H(X) = C'$, we have

$$\frac{C'}{B} = \frac{R_b}{B}$$

which leads to

$$\frac{R_b}{B} = \log_2\left(\frac{E_bR_b}{N_0B}+1\right)$$

which provides a relationship between the signal to noise ration E_b/N_0 and the modulation **efficiency** R_b/B (measured in bits/second per hertz). In particular, we can write

$$\frac{E_b}{N_0} = \frac{2^{R_b/B}-1}{R_b/B}$$

- If $\frac{R_b}{B} = 1$, then $\frac{E_b}{N_0} = 1\ (0\ dB)$;
- if $\frac{R_b}{B} \to 0$, $\frac{E_b}{N_0} \to \infty$;
- if $\frac{R_b}{B} \to \infty$, then $\frac{E_b}{N_0} \to \ln 2\ (-1.6\ dB)$:

$$\lim_{R_b/B\to\infty} \frac{2^{R_b/B}-1}{R_b/B} = \lim_{R_b/B\to\infty} \frac{e^{R_b/B\log_e 2}-1}{R_b/B} = \lim_{R_b/B\to\infty} \frac{1-R_b/B\log_e 2-1}{R_b/B} = \ln 2$$
$$= 0.693$$

The last limit is quite interesting: it starts that it is possible to transmit with error probability equal to zero if the signal to noise ratio is $E_b/N_0 > -1.6\,dB$, provided that the bandwidth B is infinite. Note that $E_b/N_0 = 1.6\,dB$ in the case in which the noise variance $N_0/2$ is equal to $0.72E_b$, really very high. Another interesting consideration is that we can trade energy with bandwidth: if we increase the bandwidth, we can reduce E_b/N_0 and vice-versa.

Note that it is not possible to get error probability equal to zero if, having fixed E_b/N_0, the spectral efficiency is higher than the value shown in the curve of Fig. 1.5; similarly it is not possible to get error probability equal to zero, if, having fixed the spectral efficiency R_b/B, the signal to noise ratio is lower than the value shown in the curve of Fig.1.5. In principle, any transmission system specified by a couple of values (E_b/N_0), (R_b/B) below the curve in Fig.1.5 can work with error probability equal to zero.

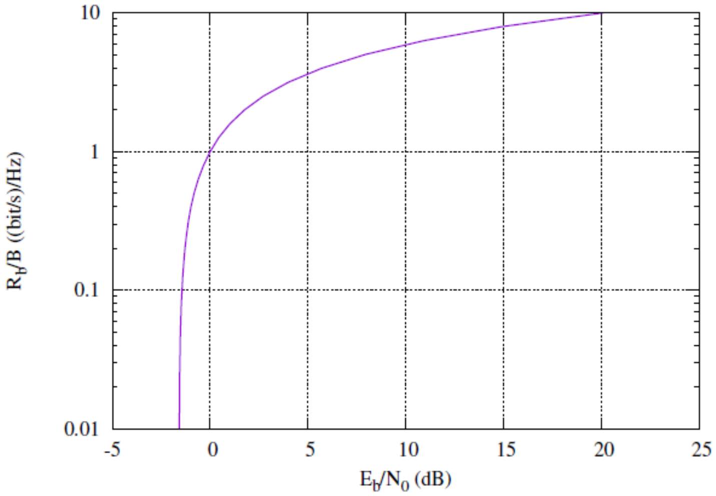


Fig.1.5: Plot Shannon channel capacity curve of spectral efficiency R_b/B versus E_b/N_0 for the AWGN channel (channel capacity limit)

In summary, the Shannon channel capacity curve, meaning the theoretical tightest upper bound on the information rate of data that can be communicated at an arbitrarily low error rate using an average received signal power through an analog communication channel subject to additive white Gaussian noise of power:

$$C' = B\log_2\left(1+\frac{P_\xi}{N_0 B}\right)$$

where

- C' is the channel capacity in information bits per second, a theoretical upper bound on the net bit rate (information rate) excluding error-correction codes;

- B is the bandwidth of the channel in hertz (passband bandwidth in case of a bandpass signal);
- P_ξ is the average received signal power over the bandwidth (in case of a carrier-modulated passband transmission), measured in watts (or volts squared);
- N_0 is the average power of the noise and interference over the bandwidth, measured in watts (or volts squared);
- $P_\xi/(N_0 B)$ is the signal-to-noise ratio (SNR) of the communication signal to the noise and interference at the receiver (expressed as a linear power ratio, not as logarithmic decibels).

1.4 Fano inequality

Consider a channel with $N_T = N_R = N$ inputs and outputs, for which it is then possible to define an error probability

$$\mathrm{P}(e) = \sum_{i=1}^{N} \sum_{j \neq i}^{N} P(y_j|x_i)\,\pi_i$$

Intuitively, $H(X|Y)$ increases if $P(e)$ increased, but Fano inequality describe the exactly relationship between capacity and the error probability. **Fano inequality** states that

$$\boldsymbol{H(X|Y) \leq \mathcal{H}\big(P(e)\big) + P(e)\,\mathbf{log_2}(N-1)}$$

being $\mathcal{H}\big(P(e)\big) = -P(e)\log_2 P(e) - P(c)\log_2 P(c)$, where $P(c)$ called "correct probability", defined by $P(c) = 1 - P(e)$.

The **intuitive proof** is the following: the receiver has observed the channel output y and it asks the Maxwell's demon the following two questions:

1. Has there been an error?
2. If there was an error, which other symbol, different from the received one, was transmitted?

The answer is to the first question is "yes" with probability $P(e)$ and "no" with probability $1 - P(e)$, so that the average quantity of information provided by the demon is $-P(e)\log_2 P(e) - \big(1 - P(e)\big)\log_2\big(1 - P(e)\big) = \mathcal{H}\big(P(e)\big)$; if the answer to the first question is yes, then the demon must answer the second question, and in that the case he provided at most $\log_2(N-1)$ information bits. It is possible that, once y is known, $\mathrm{P}(x_k|y)$ is equal to $1/(N-1)$ for $x_k \neq y$, and in this case the demon is like a source with $N-1$ eqaully likely symbols, thus providing $\log_2(N-1)$ information bits. But it is also possible (and in general this is the case), that some probabilities $\mathrm{P}(x_k|y)$ is higher than others, in which case the demon behaves like a source with an entropy smaller than $\log_2(N-1)$. Then, in the overall, the average information quantity provided by the demon is lower than $\mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$ (note that the demon provides the answer to the second question only with probability $P(e)$. At the end of the game with demon, the receiver exactly knows which symbol has been transmitted, and therefore all its uncertainty $H(X|Y)$ has been removed. Therefore, the uncertainty $H(X|Y)$ of the receiver is lower than $\mathcal{H}\big(P(e)\big)+P(e)\log_2(N-1)$.

The formal proof is instead following:

$$D = H(X|Y) - \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$$

$$= -\sum_i\sum_j P(x_i,y_j)\log_2 P(x_i|y_j) + P(e)\log_2 P(e)$$

$$+ P(C)\log_2 P(C) - P(e)\log_2(N-1)$$

$$= -\sum_i\sum_{j\neq i} P(x_i,y_j)\log_2 P(x_i|y_j) - \sum_i P(x_i,y_j)\log_2 P(x_i|y_j)$$

$$+ P(e)\log_2\frac{P(e)}{N-1} + P(C)\log_2 P(C)$$

$$= \sum_i\sum_{j\neq i} P(x_i,y_j)\log_2\frac{P(e)}{(N-1)P(x_i|y_j)}$$

$$+ \sum_i P(x_i,y_j)\log_2\frac{P(C)}{P(x_i|y_j)}$$

$$\leq \frac{1}{\log_e 2}\left\{\sum_i\sum_{j\neq i} P(x_i,y_j)\left(\log_2\frac{P(e)}{(N-1)P(x_i|y_j)} - 1\right)\right.$$

$$\left. + \sum_i P(x_i,y_j)\log_2\frac{P(C)}{P(x_i|y_j)}\right\}$$

$$= \frac{1}{\log_e 2}\left\{\sum_i\sum_{j\neq i}\frac{P(e)P(y_j)}{N-1} - \sum_i\sum_{j\neq i} P(x_i,y_j) + \sum_i P(C)P(y_i)\right.$$

$$\left. - \sum_i P(x_i,y_i)\right\} = \frac{1}{\log_e 2}\left\{\frac{P(e)}{N-1}\sum_i\sum_{j\neq i} P(y_j) - 1 + P(C)\right\}$$

$$= \frac{1}{\log_e 2}\left\{\frac{P(e)}{N-1}\left[\sum_i\sum_j P(y_j) - \sum_i P(y_i)\right] - 1 + P(C)\right\}$$

$$= \frac{1}{\log_e 2}\left\{\frac{P(e)}{N-1}[N-1] - 1 + P(C)\right\} = \frac{1}{\log_e 2}\{P(e) - 1 + P(C)\}$$

$$= 0$$

1.5 Relationship among P(e), C and H(X)

We can further process Fano inequality to obtain a relationship among C, $H(X)$ and $P(e)$. In fact:

$$C \geq I(X;Y) = H(X) - H(X|Y) \geq H(X) - \mathcal{H}\big(P(e)\big) - P(e)log_2(N-1)$$

Therefore, we get

$$H(X) \leq C + \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$$

which is always true, for any channel.

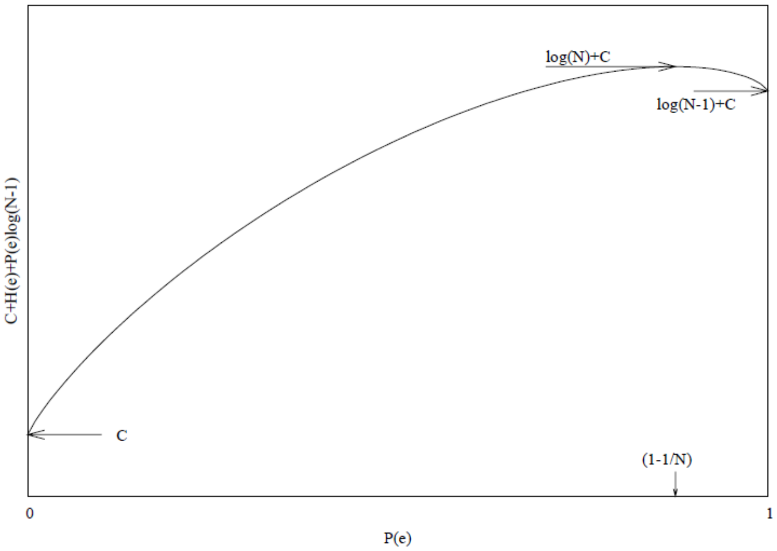


Figure 1.6: Plot of $\mathrm{C} + \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$

If symbols x_i are equally likely, then $H(X) = H(Y) = \log_2 N$, and $H(X|Y) = H(Y|X) = \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$

Figure 1.6 shows the plot of $C + \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$. Any transmission system has $H(X)$ below the curve in figure 1.6: this means that it is not possible that a system exists with $H(X) > C$ and $P(e) = 0$, while it is possible that systems exist with $H(X) < C$ and $P(e) = 0$. If the input of the channel is $H(X) > C$, then it is as if we had a pipe with more input water flow than allowed by the pipe capacity: the pipe breaks. Similarly the discrete channel "breaks" and the error probability is strictly larger than zero. However, the inequality does not say how we can get $P(e) = 0$ setting $H(X) < C$.

1.6 The converse of the channel coding theorem

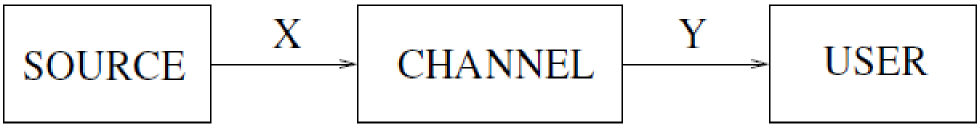


Figure 1.6: Transmission system without channel coding

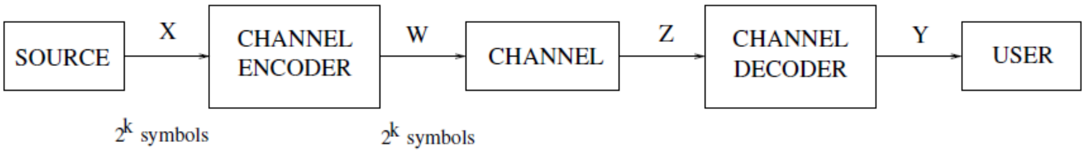


Fig. 1.7: Transmission system with channel encoding

The discussion, so far, considered the source of Fig. 1.7. Let us instead analyze the system of Fig. 1.7, where a channel encoder is placed between the source and the channel. The source generates word of k bits, so that the source alphabet X has 2^k symbols. The channel encoder maps k input bits to n output bits, the entropy of k input bits is $H(X)$ and the entropy of the n bits at the output of the encoder is still $H(X)$, since the encoder does not add information. In order to be able to distinguish the various sections of the system, let the output of the channel encoder be denoted with W (n bits per symbol). The channel is used n times, so its capacity is nC, being C its capacity per use. The channel output Z (n bits) is processed by the channel decoder which outputs k bits, identified as Y; note that the channel encoder and decoders do not make errors in their mapping processes. Then the inequality

$$H(X|Y) \leq \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$$

is valid for the outer section, $(X - Y)$, but $P(e)$ is now the probability that a word of k bits is not correctly received (it is sufficient that one bit is wrong), and it is wise to identify it as $P_w(e)$; moreover the number of symbols of source X is 2^k (all the possible sequences of k bits). These considerations lead to the more practical inequality, which based on the complete transmission system figure 1.7:

$$H(X|Y) \leq \mathcal{H}(P_w(e)) + P_w(e)\log_2(2^k - 1)$$

Using the data processing theorem, we can say that $I(X;Y) \leq I(W;Z)$ and this means that

$$I(X;Y) = H(X) - H(X|Y) \leq I(W;Z) \leq nC$$

i.e. the mutual information of X and Y is still upper bound by nC. In particular, from the last inequality we get

$$H(X|Y) \geq H(X) - nC$$

Then, combing $H(X|Y) \leq \mathcal{H}\big(P(e)\big) + P(e)\log_2(N-1)$ and $H(X|Y) \geq H(X) - nC$, we get the overall chain of inequalities

$$H(X) - nC \leq \mathcal{H}\big(P_w(e)\big) + P_w(e)\log_2(2^k - 1) < 1 + kP_w(e)$$

which allows to find a bound for $P_w(e)$:

$$P_w(e) \geq \frac{H(X) - nC - 1}{k}$$

In fact, the source generates one bit at a time and the bits are grouped into words of k bits. Then, if the entropy of the single bit generated by the source, taking into consideration also the source memory, is $H(X_1)$ (between 0 and 1, and $H(X_1) = 1$ only if the bits “0” and “1” are generated with equal probabilities by a memoryless source), then the entropy of the word source X is $H(X) = kH(X_1)$. This allows us to further simplify the inequality:

$$P_w(e) \geq \frac{kH(X_1) - nC - 1}{k} = H(X_1) - \frac{n}{k}\mathrm{C} - \frac{1}{k}$$

Being $H(X_1)$ a property of the source (which cannot be modified), the inequality shows that it is convenient to use large values of $\frac{n}{k}$, but certainly, if the right hand side of inequality is less than zero, the inequality obviously becomes $P_w(e) \geq 0$. The theorem is interesting because it gives a lower bound on the error probability and, therefore, we can say that as far as $H(X_1) - \frac{n}{k}\mathrm{C} - \frac{1}{k}$ is positive, then the error probability will be positive. But, a lower bound on the error probability is more useful, and this lower bound is provided by the channel coding theorem in the next section.

1.7 The channel coding theorem (Shannon 1948)

THEOREM (Shannon 1948):

Given a binary information source with entropy $H(X)$ and a discrete, memoryless, channel with capacity C, there exist a channel encoder with rate (also called coding rate) $R_c = \frac{k}{n}$ for which the word error probability $P_w(e)$ is upper bounded by

$$P_w(e) < e^{-nE(R)}, \qquad R = R_c H(X) = \frac{k}{n} H(X)$$

where $E(R)$ is a convex, decreasing, positive function of R for $0 \leq R \leq C$ and $E(R) = 0$ for $R \geq C$.

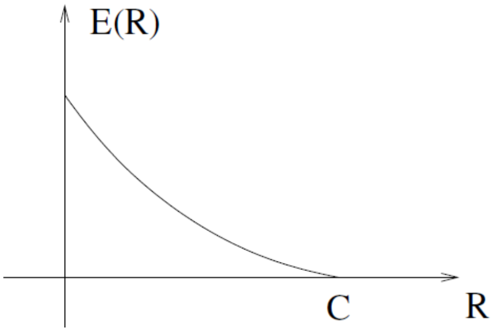


Fig.1.8: typical behavior of function $E(R)$

Fig. 1.8 shows an example of function $E(R)$. Let us assume, for simplicity and without loss of generality, that $H(X) = 1$ information bit (we assume that an ideal source encoder has been used, or that a memoryless source naturally generates equal likely bits), so that $R = R_C = k/n$. Then the channel coding theorem shows that it is possible to reduce the word error probability, for a given channel with capacity C (i.e. for a given function $E(R)$),

- If the coding rate k/n is decreased
- Or, more important, if k/n is kept constant, but n is increased (k is also proportionally increased to keep the ratio constant)

Otherwise, it is possible to reduce $P_w(e)$ by increasing the capacity C, which is typically obtained by increasing the transmitted power, so that the signal to noise ratio E_b/N_0 is increased. The increase of the capacity from $C$to $C' > C$ corresponds to a new curve $E'(R) > E(R)$ and therefore $P_w(e)$ decreases. This last solution to decrease the error probability is obvious, it works also if $k = n = 1$, and it is not necessary to use the channel coding theorem to understand it, it is sufficient to plot the error probability of the selected modulation scheme versus E_b/N_0.

Actually, the unexpected result of the channel coding theorem is that you can reduce the error probability by increasing the length k of the word (and the length n of the code word), keeping $k = n$ fixed. Actually, by increasing k, we increase the complexity of the system, and therefore we get a reduced $P_w(e)$ at the cost of an increased cost of the system. In some cases, the complexity can be so high, that practical, real time, implementation of the system is unfeasible. Great improvements have been obtained in the recent years in the design of channel coding systems which can effectively reduce $P_w(e)$ keeping the complexity limited and affordable.

A very important consideration is necessary: in the previous analysis (derivation of the coding theorem and its inverse), we always considered the channel capacity C of the channel fixed, while we let the two parameters k and n vary.

Assume that, in the absence of channel encoding, the channel transmits R_C bits per second and that the source generates a total of M bits at speed R_b bits/s, so that the time required to transmit the generated bits is M/R_b. The channel bit rate R_C in this scenario is equal to the source bit rate R_b.

Actually, there are two different scenarios when we include the channel encoder with coding rate $k/n < 1$:

1. The channel bit rate R_C is not modified, while the source bit rate R_b is reduced to $R_b{}'$; the time required to transmit $\frac{Mn}{k}$ channel bits is $(n/k)MR_c > MR_b$: we need more time to transmit the data, since the channel encoder generates $n - k$

redundancy bits every k input bits, and the redundancy bits must be transmitted over the channel; of course $R_b{}' = R_b\, k/n$.

2. the source bit rate R_b is not modified, which means that the channel now must transmit n bits instead of k bits in the same amount of time k/R_b ; this is possible only if the channel bit rate is modified, using $R_c{}' = R_c\, n/k$ instead of R_c . The increase of the channel bit rate can be obtained in two ways:
    1. The modulation scheme is not modified, but, since the encoded bits arrive at the input of the modulator with a higher rate, the duration of the waveforms generated by the modulator is reduced and the modulated signal bandwidth is therefore increased by a factor n/k. Then we have two possibilities:
        i. We increase the modulated signal power from P to $P' = \frac{P_n\, n}{k}$, so that the energy of the modulated symbol $E = \frac{P}{R_c} = \frac{P'}{R_c{}'}$
        ii. We do not modify the power P, but this means that the energy of the modulated symbol reduces from E to $\frac{Ek}{n}$ when the channel encoder is included in the transmission system
    2. a new modulation scheme is used (for example we compare a 4PSK modulation without channel encoding and an 8PSK modulation with channel encoding and rate $\frac{k}{n} = \frac{2}{3}$)

The coding theorem and its converse are valid only for cases 1 and 2(a)i, for which it is true that the capacity of the channel is not modified when the channel encoding scheme is included: in these two cases, the signal to noise ratio E/N_0 is not changed, but:

- In the first case the reduction in $P_w(e)$ is paid in terms of increased duration of the transmission: this solution is allowed only in case of non-real time applications (we have to send an e-mail, a file, etc.)
- In the second case the reduction of $P_w(e)$ is paid in terms of increased bandwidth and transmitted power, which is not always possible (you must typically change the amplifier, change the filters, pay for the excess bandwidth, etc.)

Case 2b is interesting because, if the new modulation scheme is adequately chosen, it is possible that we can reduce $P_w(e)$ without increasing the bandwidth and without increasing the power, thus paying the reduced error probability just in terms of complexity. The design of the channel encoders for this case is quite complex. Note, in any case, that we keep all the main transmission parameters (power, bandwidth, source bit rate) unvaried, but we vary the channel and its capacity.

Case 2(a) is the basis on the analysis of channel encoding schemes. In this case the channel matrix structure does not change but the conditional error probabilities in the matrix change and the channel capacity (per use of the channel) actually reduces when the channel encoder is included in the transmission system. The challenge is therefore that of finding a channel encoding scheme that allows us to reduce $P_w(e)$ even if the channel capacity (per use) is reduced.

Chapter 2

2 Applications

George Gamow pointed out that the application of Shannon's information theory breaks genetics and molecular biology out of the descriptive mode in to the quantitative mode, and Dr. Yockey develops this theme, discussing how information theory and coding theory can be applied to molecular biology.

The genetic information system is segregated, linear and digital. It is astonishing that the technology of information theory and coding theory has been placed in biology at least 3850 million years (Mojzsis, S.J, Kishnamurthy, Arrhenius, G., 1998. Before RNA and after: geological and geochemical constrains on molecular evolution 1-47. In: Gesteland, R.F. (Ed.), The RNA World: The nature of Modern RNA suggests a prebbiobic RNA, second ed. Cold Spring Harbor Laboratory Press, Boca Raton, FL). The genetic code performs a mapping between the sequences of the four nucleotides of DNA and mRNA to the sequences of the 20 amino acids in protein. It is highly relevant to the origin of life that the genetic code is constructed to confront and solve the problems of communication and recoding by the same principles found both in the genetic information system and in modern computer and communication codes (Hubert P. Yockey *, 1999. Origin of life on earth and Shannon's theory of communication, 1507 Balmoral Drive, Bel Air, MD 21014-5638, USA).

2.1 Coding Theory and its applications

Reliable information exchange and processing is a crucial need in biological systems. Without such reliability, living beings do not have much chance of survival. In artificial information processing systems, coding methods play an important role to guarantee the required reliability. In many cases, mutation due to chemical agents or radiation results in certain diseases, like cancer, and is also responsible for aging. DNA is replicated several million times in a lifetime of a species and if there were no error correction mechanism, the accumulation of errors during its lifetime and, on a larger scale, over millions of years of evolution would simply make genetic communication, and hence life is impossible.

In following, we will discuss some applications of coding theory in postal service in United States and molecular biology. These applications contains analyzing error correcting methods in biological system, as well as using the code theoretical models to implement various mechanisms in living systems.

Firstly, we introduce the coding theory: coding is the technique which allows improving performances of system in terms of power, error rate, range, bit rate etc. The coding technique also affects the bit rate, latency and complexity. As for the concrete strategy, given the binary information sequence to be transmitted, coding adds redundancy. The **redundancy** is the exploited at receiver side to **correct** and **detect** errors introduced by channel. There exists a formula to express the relationship between the information, redundancy and code word, unit represented as bit:

$$k + r = n$$

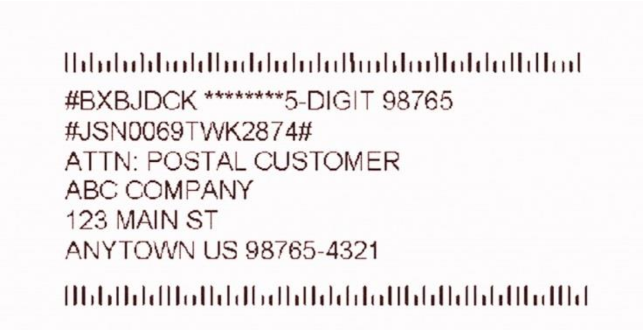
where

- k: the number of information bits;
- r: the number of redundancy bits;
- n: the number of code word bits, exactly transmitted amount;

The redundancy bits are also called no-sense code, because they have no meaning assignment in the receiving alphabet. The information bits are called sense code to assign the information meaning. The combination of sense and no-sense code can defense the noisy channel to realize error correction and direction.

2.1.1 postal ZIP+4 code

In 1983, the United States Postal Service (USPS) changed its ZIP Code system to include the new ZIP+4. A ZIP+4 Code uses the basic five-digit code plus four additional digits to identify a small delivery segment such as a street, a city block, a group of apartments, or even an individual address that receives a high volume of mail. The "ZIP" stands for "Zone Improvement Plan". A postal ZIP code contains multiple items of information compressed into nine digits. The last 4 digits of a nine-digit ZIP Code, which called an error-detecting code. For the complete, nine-digit ZIP Code, it consists of two parts. The first five digits indicate the destination the national area, the region or city, and the delivery area or post office. The last 4 digits identify a specific delivery route or a post office box, within that overall delivery area. For example, in the fig. 2.1, the first five bits “98765” is five information bits, carrying the information of destination. The last four bits “4321” is distinguished by postal employee.



#BXBJDCK ********5-DIGIT 98765
#JSN0069TWK2874#
ATTN: POSTAL CUSTOMER
ABC COMPANY
123 MAIN ST
ANYTOWN US 98765-4321

Fig.2.1An example of ZIP+4 postal code in United States

Thus the source probability space of postal codes is binary source alphabet, with 32 (2^5=32) members. The source alphabet is mapped to the receiver alphabet. In fact, the USPS has segmented the country into 10 ZIP Code areas. Starting in the northeast, they are numbered 0-9. Thus, the receiving alphabet is extracted by 10 members, from 0 to 9. Thus ten members selected by five-bits alphabet are called sense code letters, carrying useful information bits. The assignment principle of sense code letters in the five-bits alphabet of the postal ZIP+4 is shown by the first row in Table 2.1.

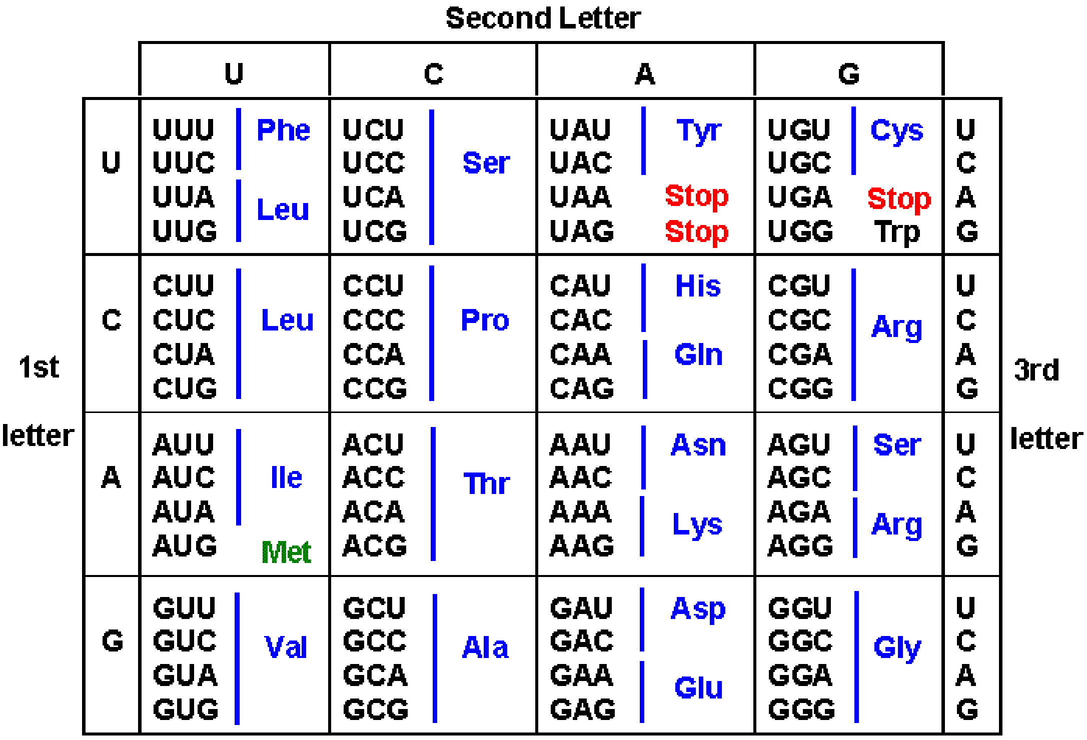
0	1	2	3	4	5	6	7	8	9
11000	00011	00101	00110	01001	01010	01100	10001	10010	10100
01000	10011	10101	10110	11001	11010	11100	00001	00010	00100
10000	01011	01101	01110	00001	00010	00100	11001	11010	11100
11100	00111	00001	00001	01101	01110	01000	10101	10110	10000
11010	00001	00111	00100	01011	01000	01110	10011	10000	10110
11001	00010	00100	00111	01000	01011	01101	10000	10001	10101

Table 2.1: The postal ZIP+4 code

In realistic life, the postal ZIP +4 code is an error-detecting code application used in United States mailing address, because a single error cannot change one sense code letter to another code letter. If the sorter reads the unintended code letters, due to dirt or other malfunctions, the postal letter or package is rejected to be checked by the postal staff.

2.1.2 The genetic code

The genetic code is a set of rules for translating a DNA or mRNA sequence with a set of three nucleotides into the amino acid sequence of a protein for protein synthesis. Almost all organisms use the same genetic code, called the standard genetic code; even non-cellular viruses use standard genetic codes. The genetic code is a block code because all codons are triplets. Each nucleotide of triplet is chosen by one of four types of nucleotides, including A, C, G, U. Thus, the source alphabet of the genetic code is the four nucleotides of DNA and mRNA. Then, there are 64 ($C_4^1 C_4^1 C_4^1 = 4 \times 4 \times 4 = 64$) possible triplets, to be translated into only 20 amino acids. The genetic code, shown in Table 2.2, shares a number of properties with the postal ZIP+4 code. Obviously, the translation process is several-to-one mapping, which the genetic code was believed to be degenerate and that some codons must be non-sense.



Second Letter

1st letter	U		C		A		G		3rd letter
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		UAA	Stop	UGA	Stop	A
	UUG		UCG		UAG	Stop	UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

Table 2.2: The mRNA genetic code

In biological systems and processes, the need to obtain and exchange information in an effective and reliable manner is one of the requirements. This information can be used to process living things and survive, just like DNA, or it can be information about the external world that is transmitted through the nervous system and processed by the nervous system. In fact, in all cases of biological systems, whether it is a single cell or a complex human central nervous system (CNS), there is a reliable and efficient mechanism for exchanging and processing information, which is a crucial requirement.

On the contrary, this medium of information exchange is not at all reliable. In the case of molecular biology, the culture medium is cytoplasm, which is an unfriendly noise environment during DNA replication, and in the case of the central nervous system, the entire pathway is noisy: the outside world, information generation, sensory systems and through it The neurons that transmit information are noisy environments.

Despite this, we enjoy a fairly accurate processing system and a fairly reliable DNA replication process. From an engineering point of view, having such a degree of reliability in the presence of noise is very alarming, and this is where coding theory comes into play.

Occurring in the cell's noisy environment, DNA replication process is not error-free. Genetic noise also occurs in all stages, during the transmission of genetic messages from the DNA to the protein tape as conceived in molecular biology, figure shown as fig.2.2. The transmission scheme is corresponding to the model of Shannon-Weaver model(1949), shown in fig.2.3.

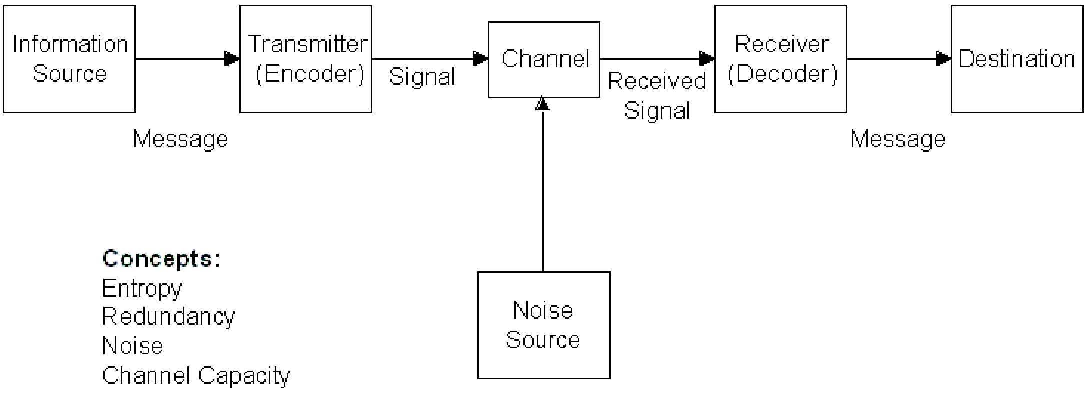


Fig. 2.2: The transmission of information from source to destination. The noise should be mentioned that occurs in all stages but is shown according to accepted practice.

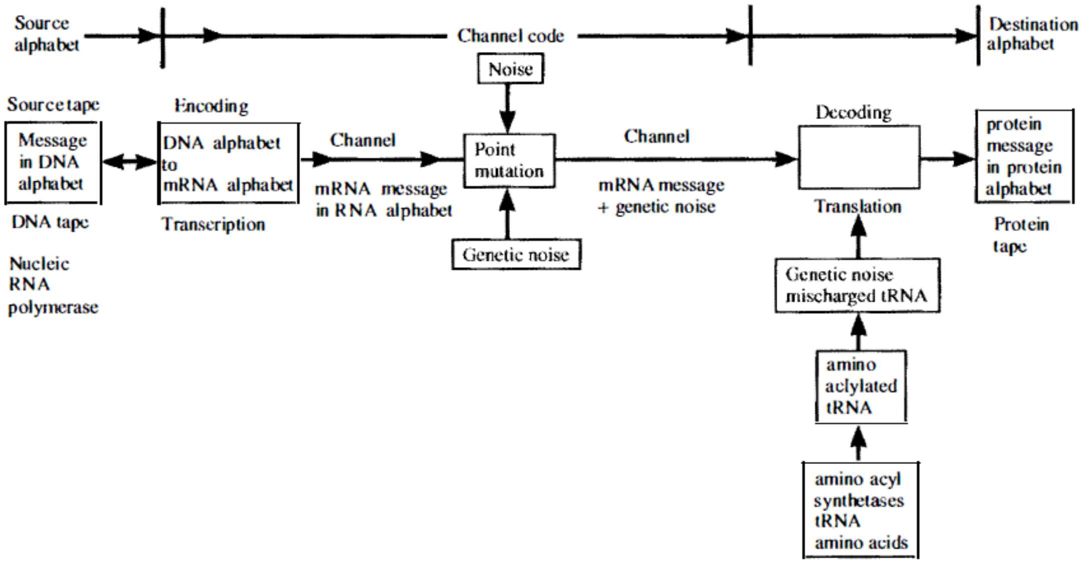


Fig. 2.3: The transmission of genetic messages from the DNA to the protein tape as conceived in molecular biology. Genetic noise occurs in all stages but is lumped in the figure to fix the idea.

There are some evidence on existence of error correction mechanisms in DNA. Firstly, as mentioned beginning of section 2.1, if there were no error correction mechanism, the accumulation of errors during its lifetime and, on a larger scale, over millions of years of evolution would simply make genetic communication, and hence life, impossible [3]. It would then be surprising to find out that the total error rate in decoding genes is as low as 10^{-9} per nucleic base and per replication. This value is noticeable as DNA replication procedure alone has an error rate of 10^{-3} to 10^{-5} [11]. One might argue that the final low error rate is a result of DNA's internal proofreading mechanism: when copied, the helical structure unzips into two separate strands. DNA polymerase uses one strand to copy DNA and the other one to check the copied segment. Although this simple proofreading reduces the error rate to approximately 10^{-6}, it is still not sufficient to explain low error probabilities observed experimentally. Therefore, in biological system, there should be error correction mechanism to resist the possible error modification of nature life.

In brief, for the biological systems, acquiring and exchanging information in an efficient and reliable manner is necessary and one of reliable needs. This information could be used to deal with the living and survival, as in DNA, or it could be information about the outside world transmitted through and processed by the neuronal system.

2.2 Shannon coding theorem and the role of error in protein function and specificity

Shannon's channel coding theorem proved that codes exist that such that sufficient redundancy can be introduced so that a message can be sent from source to receiver with as a few errors as may be specified. Error detecting and correction codes are formed by going to higher extensions and using the redundancy of the extended symbols for error detection and correction, as mentioned in section 2.1.

Note that in any case, the capability of any error correction mechanism is limited and if the number of errors increases a certain threshold (denoted as the code's minimum distance) the errors could not be detected or corrected. Therefore, mutation could be viewed as uncorrected errors in this regards. Taking into account that mutation is necessary for natural. The redundancy just provides some protection from mutation error. That is why discuss the capability of correcting error and Shannon channel coding theory. Since Shannon coding theorem describes the maximum possible efficiency of error-correcting methods versus levels of noise interference and data corruption, which shows how to compute a channel capacity from a statistical description of a channel.

Since Shannon regarded the generation of a message to be a Markov process, it was natural to measure the effect of transfer errors due to the noise by the conditional entropy $H(X|Y)$, mentioned at section 1.2.2, between the source alphabet of the genetic code with the four nucleotides of DNA and mRNA, and the receiving 20 letters alphabet of protein.

The following proof uses the same definition mentioned at section 1.2, i.e. The discrete channels are characterized by an input alphabet $X = \{x_1, x_2, \dots, x_{N_T}\}$ and an output alphabet $Y = \{y_1, y_2, \dots, y_{N_R}\}$, corresponding to the discrete channel matrix and it is possible to define the entropies for these two alphabets. With the following definitions:

$$\pi_i = P(x = x_i) \qquad \rho_j = P(y = y_j)$$

The probability $P(x_i) = \pi_i$ in source alphabet and probability $P(y_j) = \rho_j$ in receiver alphabet, are related by the following equation:

$$\rho_j = \sum_{i=1}^{N_T} P(x_i|y_j)\,\pi_i$$

The conditional entropy, $H(X|Y)$ is written in terms of the components of π_i or ρ_j, and the elements of channel matrix $\boldsymbol{P}$:

$$H(X|Y) = \sum_{i=1}^{N_T} P(x_i)\,P(x_i|y_j)\log_2 P(x_i|y_i)$$

The mutual entropy of the input and output is:

$$I(X;Y) = H(X) - H(X|Y)$$

It proves more convenient to deal with the message at the source and therefore with the π_i and the $P(y_j|x_i)$ (Yockey, 1974, 1992). Based on Bayes' theorem, conditional probabilities , we have(Feller, 1 968; Hamming, 1 986; Lindley, 1 965):

$$P(x_i|y_j) = \pi_i P(y_j|x_1)/\rho_j$$

Substituting this expression for $P(x_i|y_j)$ in equation

$$H(X|Y) = -\sum_{i=1}^{N_T} P(x_i)\,P(x_i|y_j)\log_2 P(x_i|y_i)$$

We have:

$$I(X;Y) = H(X) - H(Y|X) - \sum_{i=1}^{N_T} P(x_i)\,P(x_i|y_j)\log_2 P(x_i|y_i)$$

Where

$$H(Y|X) = -\sum \pi_i P\left(y_j\middle|x_i\right) \log_2 P\left(y_j\middle|x_i\right)$$

The third term in

$$I(X;Y) = H(X) - H(Y|X) - \sum_{i=1}^{N_T} P(x_i)\, P(x_i|y_j) \log_2 P(x_i|y_i)$$

is the information that cannot be transmitted to the receiver if the source alphabet is larger than the alphabet at the receiver, that is, if the Shannon entropy of the source is greater than that of the receiver so that the source and the receiver alphabets are not isomorphic.

For illustration we may set all the matrix elements of $\boldsymbol{P}$, $P(y_j|x_i)$ to the value given in Table 2.3 where α is the probability of misreading one nucleotide. Substituting these matrix elements in equations:

$$I(X;Y) = H(X) - H(Y|X) - \sum_{i=1}^{N_T} P(x_i)\, P(x_i|y_j) \log_2 P(x_i|y_i)$$

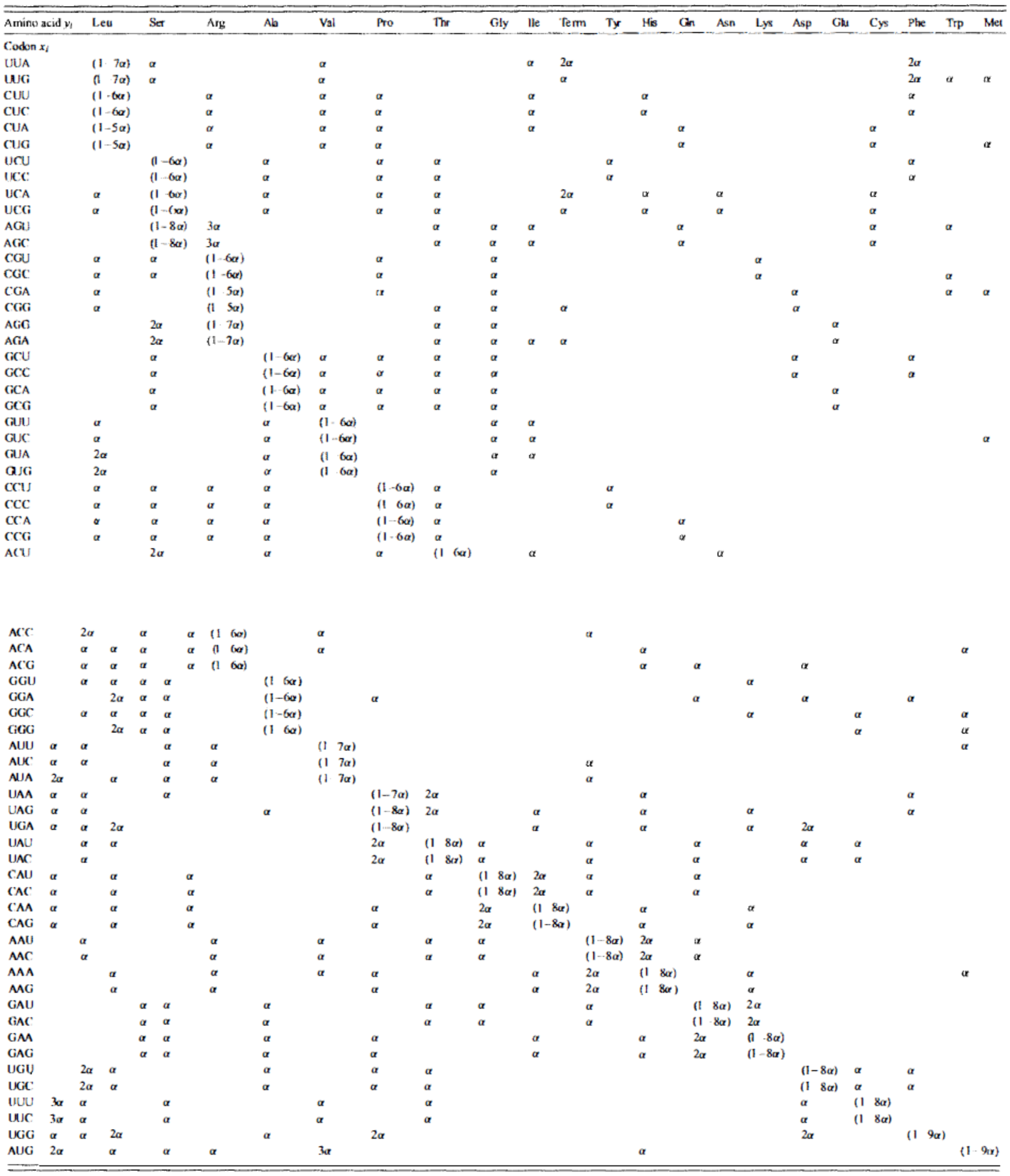
And

$$H(Y|X) = -\sum \pi_i P\left(y_j\middle|x_i\right) \log_2 P\left(y_j\middle|x_i\right)$$

And replacing the logarithm by its expansion, keeping only terms of second degree we have:

$$I(X;Y) = H(X) - 1.7915 + 34.2018\alpha^2 + 6.803\alpha\log_2\alpha$$

The genetic code cannot transfer 1.7915 bits of its six-bit alphabet to the protein sequence even without errors.



Amino acid y_j	Leu	Ser	Arg	Ala	Val	Pro	Thr	Gly	Ile	Term	Tyr	His	Gln	Asn	Lys	Asp	Glu	Cys	Phe	Trp	Met
Codon x_i																					
UUA	$(1-7\alpha)$	α			α				α	2α									2α		
UUG	$(1-7\alpha)$	α			α					α									2α	α	α
CUU	$(1-6\alpha)$		α		α	α			α			α							α		
CUC	$(1-6\alpha)$		α		α	α			α			α							α		
CUA	$(1-5\alpha)$		α		α	α			α				α					α			
CUG	$(1-5\alpha)$		α		α	α							α					α			α
UCU		$(1-6\alpha)$		α		α	α				α								α		
UCC		$(1-6\alpha)$		α		α	α				α								α		
UCA	α	$(1-6\alpha)$		α		α	α			2α		α		α				α			
UCG	α	$(1-6\alpha)$		α		α	α			α		α		α				α			
AGU		$(1-8\alpha)$	3α				α	α	α				α					α		α	
AGC		$(1-8\alpha)$	3α				α	α	α				α					α			
CGU	α	α	$(1-6\alpha)$			α		α							α						
CGC	α	α	$(1-6\alpha)$			α		α							α					α	
CGA	α		$(1-5\alpha)$			α		α								α				α	α
CGG	α		$(1-5\alpha)$				α	α		α						α					
AGG		2α	$(1-7\alpha)$				α	α									α				
AGA		2α	$(1-7\alpha)$				α	α	α	α							α				
GCU		α		$(1-6\alpha)$	α	α	α	α								α			α		
GCC		α		$(1-6\alpha)$	α	α	α	α								α			α		
GCA		α		$(1-6\alpha)$	α	α	α	α									α				
GCG		α		$(1-6\alpha)$	α	α	α	α									α				
GUU	α			α	$(1-6\alpha)$			α	α												
GUC	α			α	$(1-6\alpha)$			α	α												α
GUA	2α			α	$(1-6\alpha)$			α	α												
GUG	2α			α	$(1-6\alpha)$			α													
CCU	α	α	α	α		$(1-6\alpha)$	α				α										
CCC	α	α	α	α		$(1-6\alpha)$	α				α										
CCA	α	α	α	α		$(1-6\alpha)$	α						α								
CCG	α	α	α	α		$(1-6\alpha)$	α						α								
ACU		2α		α		α	$(1-6\alpha)$		α					α							
ACC		2α		α		α	$(1-6\alpha)$		α					α							
ACA		α	α	α		α	$(1-6\alpha)$		α						α						α
ACG		α	α	α		α	$(1-6\alpha)$								α	α		α			
GGU		α	α	α	α			$(1-6\alpha)$									α				
GGA			2α	α	α			$(1-6\alpha)$		α						α		α		α	
GGC		α	α	α	α			$(1-6\alpha)$									α		α		α
GGG			2α	α	α			$(1-6\alpha)$											α		α
AUU	α	α			α		α		$(1-7\alpha)$												α
AUC	α	α			α		α		$(1-7\alpha)$					α							
AUA	2α		α		α		α		$(1-7\alpha)$					α							
UAA	α	α			α					$(1-7\alpha)$	2α				α					α	
UAG	α	α						α		$(1-8\alpha)$	2α		α		α		α			α	
UGA	α	α	2α							$(1-8\alpha)$			α		α		α	2α			
UAU		α	α							2α	$(1-8\alpha)$	α		α		α		α	α		
UAC		α								2α	$(1-8\alpha)$	α		α		α		α	α		
CAU	α		α			α					α	$(1-8\alpha)$	2α	α		α					
CAC	α		α			α					α	$(1-8\alpha)$	2α	α		α					
CAA	α		α			α				α		2α	$(1-8\alpha)$		α		α				
CAG	α		α			α				α		2α	$(1-8\alpha)$		α		α				
AAU		α					α		α		α	α		$(1-8\alpha)$	2α	α					
AAC		α					α		α		α	α		$(1-8\alpha)$	2α	α					
AAA			α				α		α	α			α	2α	$(1-8\alpha)$		α				α
AAG			α				α			α			α	2α	$(1-8\alpha)$		α				
GAU				α	α			α			α	α		α		$(1-8\alpha)$	2α				
GAC				α	α			α			α	α		α		$(1-8\alpha)$	2α				
GAA				α	α			α		α			α		α	2α	$(1-8\alpha)$				
GAG				α	α			α		α			α		α	2α	$(1-8\alpha)$				
UGU		2α	α					α		α	α							$(1-8\alpha)$	α	α	
UGC		2α	α					α		α	α							$(1-8\alpha)$	α	α	
UUU	3α	α			α				α		α							α	$(1-8\alpha)$		
UUC	3α	α			α				α		α							α	$(1-8\alpha)$		
UGG	α	α	2α					α		2α								2α		$(1-9\alpha)$	
AUG	2α		α		α		α		3α						α						$(1-9\alpha)$

Table 2.3: Genetic code probability matrix elements $P(y_j|x_1)$

Just as some error can be tolerated in human languages, some error can be tolerated in the process of protein formation. Specific protein molecules having amino acids that differ from those coded for in DNA may have full specificity if the mutation is to a functionally equivalent amino acid. It is only when the supply of essential proteins decays below a critical level that protein error becomes lethal. Eigen and Eigen and Schuster, find an “error threshold” to apply to the transfer of information from DNA through mRNA to protein. When calculated correctly, there is no “error catastrophe”.

2. 3 Why life cannot be “protein first”

The central dogma of molecular biology is an explanation of the flow of genetic information within a biological system. It is often stated as "DNA makes RNA and RNA makes protein," although this is an oversimplification. It was first stated by Francis Crick in 1958:

“ The Central Dogma. This states that once 'information' has passed into protein it cannot get out again. In more detail, the transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible. Information means here the precise determination of sequence, either of bases in the nucleic acid or of amino acid residues in the protein. ”

—Francis Crick, 1958

and re-stated in a Nature paper published in 1970:

“ The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred back from protein to either protein or nucleic acid. ”

—Francis Crick, 1970

The Central Dogma molecular biology suggested that information could flow from DNA to RNA and from RNA to protein, but not from protein to DNA or mRNA or from protein to protein. The questions emerged among molecular biology of how a four letter alphabet could send information to a 20 letters alphabet of protein. In the full glory of its mathematical generality it applies to all codes where the information entropy of source alphabet is larger than that of the receiver alphabet.

It is obvious that if the source and receiver alphabets have the same number of symbols, and one to one correspondence between the members of the alphabets, the logic operation has a single valued inverse and information may be passed, without loss, in either direction. Thus since RNA and DNA both have a four letters alphabet, genetic message may be passed from RNA to DNA. The passage of information from protein to RNA or DNA is prohibited for three reasons, which speculates that life cannot be “protein first”:

1. The logic OR gate is irreversible
2. There is not enough information in the 20 letter protein alphabet to determine the 64 letter mRNA alphabet from which it is translated.
3. Kolmogorov proved that two sequences are not isomorphic unless they have the same entropy. The entropy of the DNA sequence is $\log_2 61$. The entropy of the protein sequence is $\log_2 20$. Obviously, these two sequences are not isomorphic. Therefore, a mapping or a several-to-one code must exist to send information

from DNA and mRNA to protein.

Conclusion

Shannon theory is invented by Claude Shannon, in the late 1940s, a mathematical theory of communication that gave the first framework in which to optimally design communication systems. The main questions motivating this were how to design communication systems to carry the maximum amount of information and how to correct for distortions on the lines. In relative, the contribution of Shannon theory is introduced the concept of information theory and information entropy, where defined a quantity of information. Otherwise, Shannon's ground-breaking approach introduced a simple abstraction of human communication, called the channel. The communication channel consisted of a transmitter (a source of information), a transmission medium (with noise and distortion), and a receiver (whose goal is to reconstruct the sender's messages).

Information entropy is described firstly in the thesis, for the most important feature of Shannon theory, which in order to quantitatively analyze transmission through the channel. Meanwhile, it introduces a measure of the average quantity of information in a message or event, which has the maximum value, described by maximum entropy theorems. In general, the more uncertain or random the message is, the more information it will contain. In terms of two source types, discrete and analog, the entropy defined by corresponding to discrete entropy and differential entropy. The differential entropy can be negative, called negative entropy. Thus, the differential entropy loses the natural property of entropy being of being positive.

In information theory, Shannon's source coding theorem (or noiseless coding theorem) establishes the limits to possible data compression, and the operational meaning of the information entropy. The source coding theorem places an upper and a lower bound on the minimal possible expected length of code words as a function of the entropy of the input word (which is viewed as a random variable) and of the size of the target alphabet.

To complete the quantitative analysis of the communication channel, Shannon introduced the entropy rate, a quantity that measured a source information production rate, and also a measure of the information carrying capacity, called the communication channel capacity.

In information theory, the other coding theorem, Shannon's noisy channel coding theorem (1948) describes the maximum possible efficiency of error-correcting methods versus levels of noise interference and data corruption, which shows how to compute a channel capacity from a statistical description of a channel. Given a noisy channel capacity and information transmitted at entropy rate, if entropy rate exceeds the channel capacity, there were unavoidable and uncorrectable errors in the transmission. In convert, there exists a coding technique which allows the probability of error at the receiver to be made arbitrarily small. This means that theoretically, it is possible to transmit information nearly without error up to nearly a limit of channel capacity bits per second.

As for the application, the initial motivation of Shannon theory is to remove the noise during communication, which gives the upper limit of the communication rate. This conclusion was firstly applied on the phone, and later applied on fiber, and now applied on the wireless communication. Today we are able to clearly take ocean telephones or satellite phones, which are closely related to the improvement of communication channel quality. In the thesis, applications extend to biology region with genetic coding.

Reference

1. Hubert P. Yockey, Information theory, evolution and the origin of life, 1507 Balmoral Drive, Bel Air, MD 21014-5638, USA
2. Mojzsis, S.J, Kishnamurthy, Arrhenius, G., 1998. Before RNA and after: geological and geochemical constrains on molecular evolution 1-47. In: Gesteland, R.F.
3. G. Battail, ”Should Genetics Get an Information-Theoretic Education? ”, IEEE Eng. Med. and Bio. Mag., Vol. 25, No. 1, 2006, pp. 34-45.
4. M. Eigen, P. Schuster, The hypercycle: a principle of natural self-organization. Part A: emergence of the hypercycle, Naturwissenschaften 64 (1977) 541–565.
5. M. Eigen, Self-organization of matter and the evolution of biological macromolecules, Naturwissenschaften 58 (1971) 465–523.
6. H.P. Yockey, An application of information theory to the Central Dogma and the sequence, J. Hypothesis Theor. Biol. 46 (1974) 369–406.
7. H.P. Yockey, Information Theory and Molecular Biology, Cambridge University Press, Cambridge, 1992.
8. H.P. Yockey, Origin of life on earth and Shannon’s theory of communication, Comput. Chem. 24 (2000) 105–123.
9. Monica Visintin, Lecture notes: Communications and Information Theory, Draft version 5, 2014
10. Amir Hesam Salavati, Algorithmics Laboratory (ALGO), I&C: Applications of Coding Theory in Biological
11. HUBERT P. YOCKEY, Information theory, evolution, and the origin of life, Cambridge University
12. G. L. Rosen, ”Examining coding structure and redundancy in DNA”, IEEE Eng. Med. and Bio. Mag., Vol. 25, No. 1, 2006, pp. 62-68.