

# Information Theory A - Class notes

Gabriel Barberini 2023/02

*How to count the degree of uncertainty that gets eliminated or the amount of information that gets introduced by a system?*

*"Information is the resolution of uncertainty." C Shannon 1948*

## Probability fundamentals

### Joint probability

$r(x_i, y_j) = r_{ij} = p(x_i, y_j)$  is the probability that experiment (X, Y) yields  $(x_i, y_j)$

$$r(x_i, y_j) = p(y_j) \cdot q(x_i/y_j) = p(x_i) \cdot q(y_j/x_i)$$

$r(x_i, y_j) = p(y_j) \cdot p(x_i)$ , when  $x_i, y_j$  are independent

---

### Marginal probability

$p_i = \sum_{j=1}^m r_{ij}$  is the probability of occurrence of  $(x_i, j)$ , that is  $p(i) = r_j$

---

### Conditional probability

$$q(y_j/x_i) = \frac{r_{ij}}{p_i}$$

where  $\sum_{j=1}^m q_{ji} = 1$  and, generally  $\sum_{i=1}^m q_{ji} \neq 1$

## Bayes theorem

$q(y_j/x_i) = \frac{p_j}{p_i} \cdot q(x_i/y_j)$ , that is, given that the conditional prob  $q_{ij}$  is known, one can determine the conditional prob  $q_{ji}$

---

## Statistical independence

$$q(y_j/x_i) = p(y_j)$$

---

## Complete statistical dependence

$$q(y_j/x_i) = 1$$

---

## Information theory fundamentals

Hartley first proposed the information measure of a given message of size  $m$  made out of a symbol set of size  $n$  as:

$$H(n^m) = \log(n^m) = m * \log(n)$$

Note that this does not take into consideration the possibility that the symbols may have unequal chances of occurring, **or that there could be a possible dependence** between the  $m$  successive symbols.

## Shannon's average amount of information

Shannon extends Hartley measure by weighting the amount of information by the probability of occurrence of each symbol:

$$H(X) = - \sum_{i=1}^n p(x_i) \cdot \log(p(x_i))$$

That is, entropy  $H(X)$  is the average amount of information needed to specify (fully distinguish) a given element  $x$  out of a set of  $X$  elements.

Where

1.  $H(P)$  is continuous on  $P$
2.  $H(P)$  is symmetric (ordering/permutation of  $p_i$  in  $P$  does not influence value of  $H(P)$ )
3.  $H(P)$  is additive, if  $X \perp Y$  then  $H(X, Y) = H(X) + H(Y)$
4.  $H(P)$  is max when  $p$  is the same for every  $p_i$  in  $P$
5.  $\log(\text{len}(P)) \geq H(P) \geq 0$

## Information and the amount of questions

The amount of information *info* (bits for  $n=2$ , nats for  $n=e$ , etc) corresponds to the **total amount of questions with  $n$  possible answers** that needs to be made to fully distinguish a point in an equal probable space  $n^{\text{info}}$

## Information example

A sample space  $X$  is given by  $X = (x_1, x_2, x_3)$ , while the accompanying probability space is given by  $P = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ .

Playing a "yes" and "no" game ( $n=2$ ), it seems obvious to ask for  $x_1$  first, as this outcome has the greatest probability. If the answer is "yes", then we have found the outcome in one go. If the answer is "no", then the outcome is obviously  $x_2$  or  $x_3$ . To determine if it is  $x_2$  or  $x_3$  costs another question, so we need to ask two questions in total to know the outcome.

**One must therefore ask either one or two questions, with equal probabilities**, hence the average is 1.5 questions. If we calculate the amount of information according to Shannon, then we find that:

$$H(X) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - 2\left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right) = 1.5 \text{ bits}$$

Joint information measure

$$H(X, Y) = - \sum_{i=1}^n \sum_{j=1}^m r(x_i, y_j) \cdot \log[r(x_i, y_j)]$$
$$H(X, Y) = H(X) + H(Y/X)$$

---

Conditional information measure

Experiment Y with regard to a given outcome:

$$H(Y/x_i) = - \sum_{j=1}^m q(y_j/x_i) \cdot \log[q(y_j/x_i)]$$

---

Experiment Y with regard to a given experiment X:

$$H(Y/X) = - \sum_{i=1}^n \sum_{j=1}^m r(x_i, y_j) \cdot \log[q(y_j/x_i)]$$
$$H(Y/X) = - \sum_{i=1}^n p(x_i) \cdot H(Y/x_i)$$

---

Experiment X with regard to a given experiment Y:

$$H(X/Y) = - \sum_{i=1}^n \sum_{j=1}^m r(x_i, y_j) \cdot \log[q(x_i/y_j)]$$
$$H(X/Y) = - \sum_{j=1}^m p(y_j) \cdot H(X/y_j)$$

---

Mutual information measure

$I(X; Y)$  can be interpreted as a measure for the dependence between  $Y$  and  $X$ .

$$I(X; Y) = - \sum_{i=1}^n \sum_{j=1}^m r(x_i, y_j) \cdot \log \left[ \frac{r(x_i, y_j)}{p(x_i)p(y_j)} \right]$$

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

When  $X$  and  $Y$  are completely independent, then  $H(X/Y) = H(X)$  and  $I(X; Y)$  is **minimum**, namely  $I(X; Y) = 0$ .

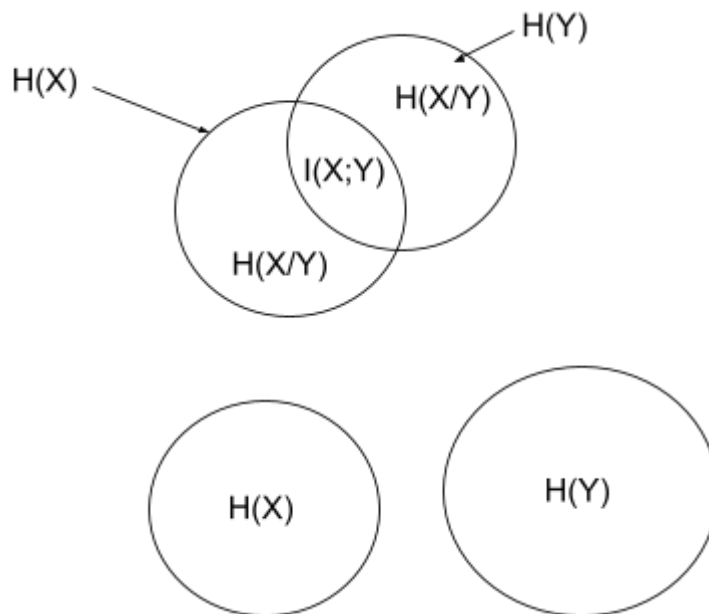
If  $Y$  is **completely dependent on**  $X$  then  $H(Y/X) = 0$  and  $I(X; Y)$  attains its **maximum value** which is equal to  $I(X; Y) = H(Y) = H(X)$

---

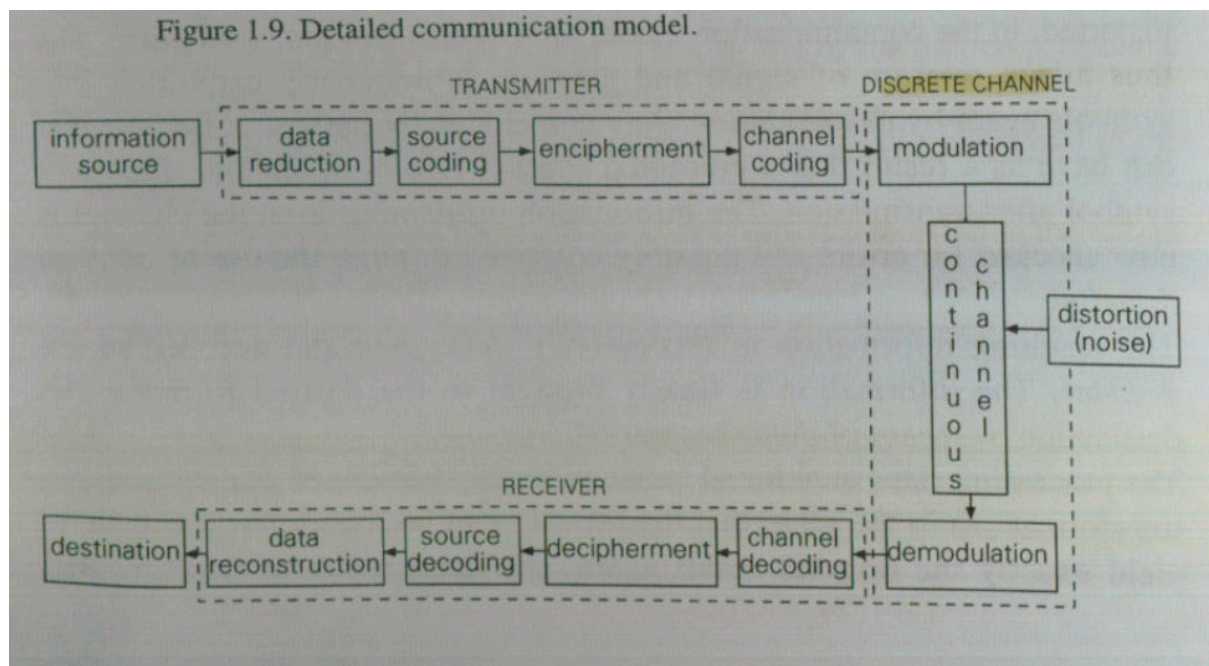
Information measures corollary

$$I(X; Y) = H(X) \cap H(Y)$$

$$H(X, Y) = H(X) \cup H(Y)$$



## Detailed communication model



## Information sources

### The discrete memoryless information source

By making a distinction between symbols (alphabet) and messages (words) we can view the information source in two ways: at the **symbolic level** and at the **message level**.

Considering the information source at the symbolic level

The amount of information that is generated by the discrete memoryless source is equal to:

$$H(U) = - \sum_{i=1}^n p_i \cdot \log_2(p_i) \text{ bits/symbol}$$

Where  $p$  is the probability of occurrence of a given symbol  $u$  from the alphabet of symbols  $U$ .

The maximum amount of information that can be generated by a discrete memoryless source is:

$$\max H(U) = \log_2(n)$$

Which is when every symbol  $u$  out of an alphabet of  $U$  symbols contains an equal probability of occurrence  $p$ .

Considering the information source at the message level

$$H(V) = - \sum_{j=1}^{n^l} p(v_j) \cdot \log_2(p(v_j))$$

Where  $v \in V$  are the possible messages of size  $l$  from an alphabet of  $n$  symbols where symbol repetition is allowed.

Assuming each symbol in the alphabet is independent, by writing  $p(v)$  (words probability) in terms of  $p(u)$  (symbols probability) we can easily conclude that:

$$H(V) = L \cdot H(U)$$

---

### Source encoding

Comes after data reduction and represents the messages as compactly as possible by removing the redundancy present within the message (lowering the entropy of the effective source).

---

### Code word

Combination of code symbols that map to a source symbol.

---

### Non-singular code

A code is nonsingular if each source symbol is mapped to a **different** non-empty code word, (mapping a collection of source symbols to code words is injective).

---

### Uniquely decodable code

Mapping of a succession of code words to source symbols remains non-singular (mapping a collection of source symbols to code words is bijective).

---

### Instantaneous or prefix-free code

Uniquely decodable, and each message symbol can be directly decoded without first looking at succeeding code symbols.

---

### Source encoding example

An information source has an alphabet with four source output symbols  $u_1, u_2, u_3, u_4$ . The code alphabet consists of the two symbols 0 and 1. Code words are made up using four different coding systems according to the following table:

	A	B	C	D
$u_1$	0	00	0	0
$u_2$	11	01	10	01
$u_3$	00	10	110	011
$u_4$	01	11	1110	0111

The four codes are all non-singular. **Code A is not uniquely decodable**, however, because the combination 0011 can be obtained from  $u_1u_1u_2$  or from  $u_3u_2$ . **The codes B, C and D are uniquely decodable:** B because all code words are equally long, so that one need only split the consecutive sequences of code words into groups of two symbols (check non-optimal encoding in appendix), C because every code word ends with a 0, which functions as a comma code and finally D because every code word begins with a 0. **Code D however is not instantaneously decodable**, because one must always wait for the first symbol of the next code word before the current code word can be decoded.

---

### Kraft Inequality

Kraft inequality examines a necessary and sufficient condition that a coding system must meet in order for it to be **instantaneously decodable**.

$$\sum_{i=1}^n r^{-L_i} \leq 1$$

Where  $r$  is the size of the code alphabet and  $L_i$  ( $i = 1, \dots, n$ ), the length of a code word  $c_i$ .

---

### Source code theorem

Consider all code words composed of symbols from the **code alphabet**  $S = \{s_1, s_2, \dots, s_r\}$  ( $r$  is the size of  $S$ ). **If Kraft's inequality holds**, then we can say that:

$$\frac{H(U)}{\log(r)} \leq L, \text{ where } H(U) \text{ is the amount of information of the source and } L \text{ is the average code word length.}$$



## Average codeword length

$L$  is the average code word length, defined by:

$$L = \sum_{i=1}^n p_i l_i$$

Where  $p_i$  denotes the probability that the  $i$ th codeword  $c_i$  is selected;  $l_i$  is the length of the  $i$ th codeword  $c_i$  and  $n$  is the total number of codewords.

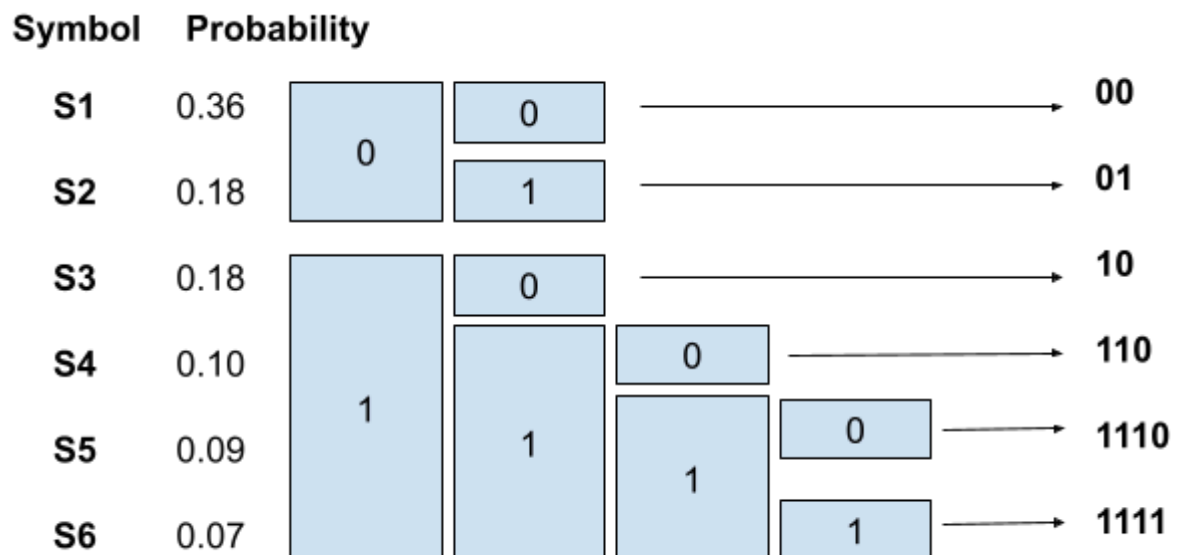
---

## Coding strategies

### Fano code

1. Order source symbols by decreasing probability;
2. Clusterize symbols as well as possible into  $n$  equally probable groups (the sum of the probability of one group should match the sum of the probabilities on the other), where  $n$  is the size of the code alphabet (2 for binary, 3 for ternary, etc);
3. Label each group with a different code symbol;
4. Repeat until no more division is possible.

e.g



### Fano code example

Symbol	Probability	Code 1 (r=2)	Code 2 (r=3)	Code 3 (r=4)
$u_1$	0.30	00	0	0
$u_2$	0.25	01	10	1
$u_3$	0.12	100	11	20
$u_4$	0.10	101	20	21
$u_5$	0.10	110	21	30
$u_6$	0.05	1110	220	31
$u_7$	0.04	11110	221	32
$u_8$	0.04	11111	222	33

### Shannon code

1. Order source symbols by decreasing probability;
2. Calculate the C.D.F (cumulative distribution function) of the source symbols in the same

$$\text{order of step 1 } P_k = \sum_{i=1}^{k-1} p(u_i) \text{ for } k \in [1, 2, \dots, n]$$

3. Write the binary (or  $n$ -ary) representation of the calculated probability in the CDF as a combination of decaying exponential of basis  $n$ :

$$\alpha n^{-1} + \beta n^{-2} + \gamma n^{-3} + \dots + \delta n^{-l_k} \mid \{\alpha, \beta, \gamma, \dots, \delta\} \subseteq S_n, \text{ given the inequality for code word length } -\log(p_k) \leq l_k < 1 - \log(p_k)$$

### Shannon code example

Symbol	Probability	$P_i$	Length $l_i$	Code (r=2)
$u_1$	$\frac{1}{4}$	$P_1 = 0$	$l_1 = 2$	00
$u_2$	$\frac{1}{4}$	$P_2 = \frac{1}{4}$	$l_2 = 2$	01
$u_3$	$\frac{1}{8}$	$P_3 = \frac{1}{2}$	$l_3 = 3$	100

$u_4$	$\frac{1}{8}$	$P_4 = \frac{5}{8}$	$l_1 = 3$	101
$u_5$	$\frac{1}{16}$	$P_5 = \frac{3}{4}$	$l_1 = 4$	1100
$u_6$	$\frac{1}{16}$	$P_6 = \frac{13}{16}$	$l_1 = 4$	1101
$u_7$	$\frac{1}{32}$	$P_7 = \frac{7}{8}$	$l_1 = 5$	11100
$u_8$	$\frac{1}{32}$	$P_8 = \frac{29}{32}$	$l_1 = 5$	11101
$u_9$	$\frac{1}{32}$	$P_9 = \frac{15}{16}$	$l_1 = 5$	11110
$u_{10}$	$\frac{1}{32}$	$P_{10} = \frac{31}{32}$	$l_1 = 5$	11111

Symbol	Probability	$P_i$	Length $l_i$	Shannon code (r=2)	Fano code (r=2)
$u_1$	0.4	0	2	00	0
$u_2$	0.3	0.4	2	01	10
$u_3$	0.2	0.7	3	101	110
$u_4$	0.1	0.9	4	1110	111

---

### Huffman code

1. In the binary (***n*-ary**) case the 2 (***n***) least probable source output symbols are joined together, resulting in a new message alphabet with one (***n*-1**) less symbol(s);
2. Step 1 is carried on until a message alphabet of just 2 (***n***) symbols arises, these two symbols are then assigned the code symbols 0 and 1 from the binary code.
3. Working backwards, a 0 or 1 is added to the code word at each place where two symbols have been joined together.

### Huffman code example

Symbol	Probability					Code (r=2)
u1	0.4	0.4	0.4	0.4	→0.6 (0)	1
u2	0.3	0.3	0.3	0.3 (0)	→0.4 (1)	00
u3	0.1	→0.1	→0.2 (0)	→0.3 (1)		011
u4	0.1	0.1 (0)	0.1 (1)			0100
u5	0.06 (0)	0.1 (1)				01010
u6	0.04 (1)					01011

OBS: Huffman code always produces instantaneous codes

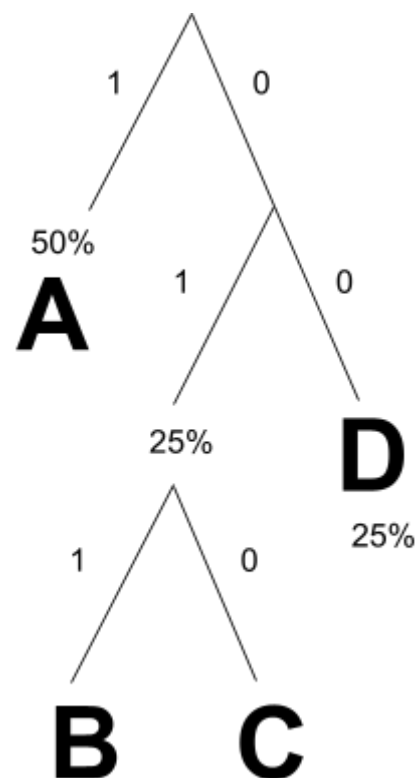
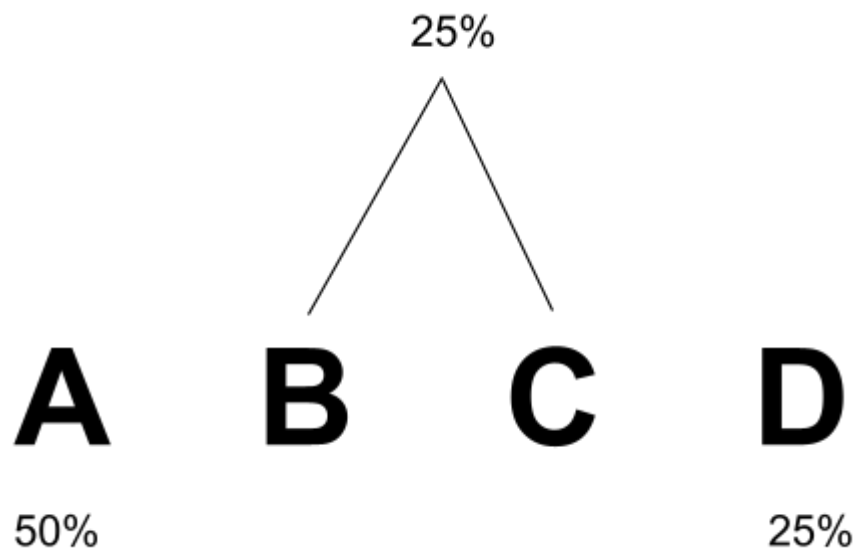
*Huffman code by graph method*

(code alphabet of size  $n$ )

1. Merge least  $n$  probable figures bottom→up, forming a node of  $P = \sum_n p$
2. Repeat 1 with the nodes and remaining figures until the tree is complete
3. Assign a unique symbol out of the code alphabet for each branch top→down
4. The code for each figure is defined by the path from the top of the tree to the figure.

**A**      **B**      **C**      **D**

50%      12,5%      12,5%      25%



**A** → **1**

**B** → **011**

**C** → **010**

**D** → **00**

---

Alphabet extension

Grouping of  $l$  source code symbols together to form a new message alphabet and subsequently use a coding strategy such as that of Huffman to obtain the eventual code.

### Alphabet extension example

Two source output symbols  $u_1$  and  $u_2$  have probabilities of  $\frac{3}{4}$  and  $\frac{1}{4}$ .

Symbol	Probability	Code (r=2)
$u_1$	$\frac{3}{4}$	0
$u_2$	$\frac{1}{4}$	1

With  $H(U) = 0.811 \text{ bit}$ ,  $L = 1$ ,  $r = 2$ ,  $\eta = 0.811$

We subsequently take two symbols together, and thus get new messages  $v_1, \dots, v_4$ . We now have:

$$H(V) = 1.622, L = \frac{27}{16}, r = 2, \eta = 0.961$$

Message	Probability	Code (r=2)
$v_1 = u_1u_1$	$p(v_1) = p(u_1u_1) = \frac{9}{16}$	0
$v_2 = u_1u_2$	$p(v_2) = p(u_1u_2) = \frac{3}{16}$	10
$v_3 = u_2u_1$	$p(v_3) = p(u_2u_1) = \frac{3}{16}$	110
$v_4 = u_2u_2$	$p(v_4) = p(u_2u_2) = \frac{1}{16}$	111

Obs: Coding has taken place on the basis of Fano's coding method.

### Finding the most probable messages

Probability  $P(\mathbf{v})$  of an arbitrary message  $\mathbf{v}$  where:

- $l_i > 0 \rightarrow$  number of times a source symbol  $u_i$  appears in a message  $v_j$
- $k \rightarrow$  number of different source symbols in the message  $\mathbf{v}$

$$p(\mathbf{v}) = \prod_{i=1}^k p(u_i)^{l_i}$$

Correlating to Shannon's entropy we have:

$$\frac{1}{l_{total}} \log p(v) \approx -H(U)$$

Where

$$l_{total} = \sum_{i=1}^k l_i$$

---

Shannon first coding theorem

Given a discrete memoryless source with information  $H(U)$  from which messages of size  $l$  are encoded into code words of size  $L$  out of a coding alphabet containing  $r$  symbols. If  $P_e$  is the probability that a message occurs for which there is no code word available. Then  $P_e$  can be made arbitrarily small as long as  $L$  satisfies:

$$L \cdot \log r \geq l \cdot H(U)$$

---

The discrete information source with memory

$P(j/i)$  is the probability that a markov chain goes over state  $S_i$  into state  $S_j$

A markov chain of order  $k$  for a stochastic variable  $u_i$  with  $m$  possible outcomes has  $m^k$  different states and  $m^{k+1}$  different transitions, each determined by a sequence of  $k$  symbols. The occurrence of a symbol is determined by the occurrence of the  $k$  preceding symbols.

Some examples:

In a degenerated Markov chain (order  $k=0$ ) there is just one state and the number of possible transitions is equal to the number of symbols that can be chosen (memoryless source).

For a Markov chain of order  $k=1$ , the number of states is equal to the number of symbols, and the number of transitions is double the number of symbols, etc.

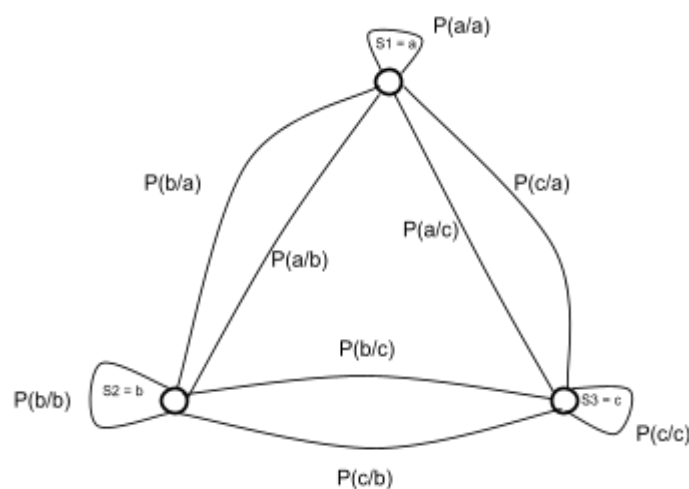
---

Marginal probability of a Markov chain state S

The probability of a given state in a markov chain can be determined from the transition probabilities:

$$P(S_i) = \sum_{j=1}^n P(S_j)P(S_i/S_j)$$

So, for a Markov chain of order  $k=1$  and 3 symbols, there are 3 states and  $3^2$  paths:



$$P(S_i) = P(S_1)P(S_i/S_1) + P(S_2)P(S_i/S_2) + P(S_3)P(S_i/S_3); i = 1,2,3$$

---

Bayes theorem and Markov chains

In Markov chains the joint probability of two states are generally not associative. Basically,  $P(S_j, S_i) \neq P(S_i, S_j)$ . Which means that Bayes theorem generally does not apply to Markov chains.

---

Markov chain properties for information theory

- The Markov chain used here is **homogeneous**. namely: stationary transition probabilities  $\rightarrow$  the transition matrix stays the same at every transition.
- The Markov chain is **ergodic**: no matter what state it finds itself in, from each state one can eventually reach the other, directly or indirectly.



Amount of information for a first-order Markov chain (k=1)

For a first-order Markov chain we have that the number of symbols  $u_i, i = 1, 2, \dots, m$ , is equal to the number of states  $S_i$

---

Amount of information of an arbitrary transition between two symbols

$$H(U_2/U_1) = - \sum_{i=1}^m \sum_{j=1}^m P(u1_i, u2_j) \cdot \log P(u2_j/u1_i)$$

---

Joint amount of information of two symbols

$$H(U_1, U_2) = - \sum_{i=1}^m \sum_{j=1}^m P(u1_i, u2_j) \cdot \log P(u1_i, u2_j)$$

$$H(U_1, U_2) = H(U_1) + H(U_2/U_1)$$

also

$$H(U_2/U_1) \leq H(U_2)$$

$$H(U_1, U_2) \leq H(U_1) + H(U_2)$$

Since the **source is stationary and ergodic**  $\rightarrow H(U_1) = H(U_2) = H(U_3)$  so

$$H(U_1, U_2) \leq 2H(U)$$

---

Amount of information for higher-order Markov chains (k>1)

The conditional amount of information of the  $N$ th symbol in the case where the  $N-1$  symbols are known is a monotonic decreasing function of  $N$ :

$$H(U_n/U_{n-1}, \dots, U_2, U_1) \leq H(U_n/U_{n-1}, \dots, U_2) \leq H(U_n/U_{n-1}) \leq H(U_n)$$

$$NH_n(U) = (N - 1)H_{n-1}(U) + F_n(U)$$

$$H(V) = H(U_1, U_2, \dots, U_n) \text{ bits/message}$$

$$H_n(U) = \frac{1}{n}H(V) \text{ bits/symbol}$$


---

Some other observations

- Probability of 0000...01  $\rightarrow P_k(0) = P(0/0)^{k-1}P(1/0)$
  - Probability of 1111...10  $\rightarrow P_k(1) = P(1/1)^{k-1}P(0/1)$
- 

## Redundancy of a discrete information source

Redundancy is a measure of how far the source is from being completely uncertain or random. When redundancy is high, there is more predictability in the source.

$$red = 1 - \frac{H(U)}{\log_2(n)}$$

Where  $H(U)$  is the amount of information (information content) of a source out of an alphabet of size  $n$ .

---

## Production of a source

$$H_t(U) = \frac{1}{t}H(U) \quad [\text{bits} / \text{s}]$$

Where  $t$  is usually the duration time of each symbol  $u \in U$  in seconds (assumed to be equal for all  $u$ ).

If the symbols do not have equal duration, as is the case with the Morse code where the "dash" takes longer than the "dot", then the average amount of time  $t$  is used instead.

## Efficiency of a code

$$\eta = \frac{H(U)}{L \cdot \log(r)}$$

Where  $H(U)$  is the amount of information of the source;  $L$  is the average code word length and  $r$  is the size of the code alphabet.

### Code efficiency example

An information source has an alphabet with four source output symbols  $u_1, u_2, u_3, u_4$ . The code alphabet consists of the two symbols 0 and 1. The probabilities of the message symbols are all  $\frac{1}{4}$ . Assume the code is as follows:

Symbol	Code ( $r = 2$ )
$u_1$	00
$u_2$	01
$u_3$	10
$u_4$	11

For this code  $H(U) = \log_2 4 = 2 \text{ bit}$ ,  $r = 2$  and  $L = 2$ . Thus the efficiency of this code is:

$$\eta = \frac{H(U)}{L \cdot \log(r)} = \frac{2}{2 \cdot 1} = 1 = 100\%$$

---

## The continuous memoryless information source

### Continuous information measure

$$H(X) = - \int_{-\infty}^{\infty} p(x) \cdot \log p(x) dx$$

## The continuous information source with memory

Joint information measure

$$H(X, Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \cdot \log p(x, y) \, dx dy$$

---

Conditional information measure

$$H(X/Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \cdot \log p(x/y) \, dx dy$$
$$H(Y/X) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \cdot \log q(y/x) \, dx dy$$

---

Mutual information measure

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \cdot \log \frac{p(x, y)}{p(x) \cdot q(y)} \, dx dy$$

If  $p(\mathbf{x}, \mathbf{y})$  follows a bi-dimensional gaussian distribution under constant power  $\sigma^2$ , we have:

$$K = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}$$

where  $K$  is the covariance matrix between  $\mathbf{x}$  and  $\mathbf{y}$

$$I(X; Y) = - \frac{1}{2} \log(1 - \rho^2)$$

So, if  $\rho = 0$  we have that  $\mathbf{x}$  and  $\mathbf{y}$  are fully independent and  $I(X; Y) = 0$ , otherwise if  $\rho = 1$  then  $\mathbf{x}$  and  $\mathbf{y}$  are totally dependent and  $I(X; Y) = H(X)$

## Redundancy of a continuous information source

Assuming limited power;  $p(x)$  follows a normal distribution under constant variance:

$$red = 1 - \frac{H(X)}{\log[\sigma\sqrt{2\pi e}]}$$

Assuming bounded amplitude;  $p(x)$  follows a uniform distribution under amplitude range of -A to +A:

$$red = 1 - \frac{H(X)}{\log(2A)}$$

---

## Information power

Let  $H(X)$  be the amount of information related to the stochastic signal  $x(t)$ . The *information power* in the case of arbitrary signals with an amount of information  $H(X)$  is:

$$P_h = \frac{1}{2\pi e} 2^{2H(X)}$$

---

## Useful integrals

$$\int x^n \ln x \, dx = \frac{x^{n+1}}{n+1} \left\{ \ln x - \frac{1}{n+1} \right\}$$

$$\int x^n \log_{\beta} x \, dx = \frac{1}{\ln \beta} \int x^n \ln x \, dx$$

---

## Communication channels

### The discrete communication channel

#### Capacity of a channel

Maximum amount of information that can be transported through the channel.

### Capacity of a discrete noiseless channel

Suppose that we have a *discrete-time* channel which is noiseless but constrained to send **one of  $M$  symbols at each channel use** (discrete time). The capacity of such a channel is, clearly  $C_d = \log_2 M$  bits *per channel use* (be careful with the capacity units).

If we can use such channel  $N$  times in a time interval  $T$ , then our capacity is  $C = \frac{N}{T} C_d = \frac{N}{T} \log_2 M$  bits per second.

For large duration  $T$  about  $N(T) = 2^{CT}$  different messages can be transmitted through the channel where the capacity of a **discrete noiseless channel with symbols of unequal duration** is given by:

$$C = \log(X_0) \text{ [bits/sec]}$$

$X_0$  being the largest positive  $X$  for which the determinant of matrix  $A$  is equal to 0:

$$[A]: a_{ij} = \sum_s X^{-t_{ij}^s} - \delta_{ij}$$

Where  $\delta_{ij}$  is the Kronecker symbol ( $\delta_{ij} = 1$  for  $i=j$ , otherwise  $\delta_{ij} = 0$ ) and  $t_{ij}^s$  the duration of a symbol  $s$  from state  $S_i$  to  $S_j$

**In other words, the capacity of a discrete channel without noise is fully limited by the entropy of the message space it supports and the duration of each message.**

---

### Capacity of a discrete noisy channel

$$C = \max_{p(y)} I(X; Y) = \max_{p(y)} \{H(Y) - H(Y/X)\}$$

[bits / symbol]

- $H(X/Y) \rightarrow$  Equivocation
  - $H(Y/X) \rightarrow$  Irrelevance (uncertainty introduced by noise)
- 

### Capacity of a discrete non-distorted channel

For a non-distorted ( $H(X/Y) = H(Y/X) = 0$ ) discrete information channel we have that  $p(x)$  is maximum when the random variable  $x$  of the input signal follows the uniform distribution, hence if such channel receives  $m$  different symbols we have:

$$C = \max_{p(x)} I(X; Y) = H(Y) = \log(m) \text{ [bits/symbol]}$$

If we assume that each symbol have a common duration of  $t$  seconds, then the channel capacity per second is given by:

$$C = \frac{1}{t} \log(m) \text{ [bits/sec]}$$

Rate of transmission

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

From this expression, the mutual information can be regarded as the difference between the uncertainty at the receiving end about the transmitted symbol  $x$  before a symbol  $y$  has been received and after a symbol  $y$  has been received.

In other words,  $I(X; Y)$  is related to the amount of information that is transported over the channel. Thus, the *rate of transmission*:

$$R = I(X; Y) \text{ bits/sec}$$

For a non-distorted channel we clearly have that  $H(X/Y) = H(Y/X) = 0$ . In that case,  $I(X; Y) = H(X)$ .

Error probability and equivocation

**For a communication channel with a square channel matrix** (number of symbols at the input of the channel is equal to the number of symbols at the output), the average error probability from the point of view of the receiver is equal to that from the point of view of the transmitter:

$$P_e = \sum_{j=1}^n q(y_j) [1 - p(x_j/y_j)]$$

$$P_e = \sum_{i=1}^n p(x_i) [1 - q(y_i/x_i)]$$

$$P_e = \sum_{i=1}^n \sum_{j=1}^n r(x_i, y_j), j \neq i$$


---

### Shannon second coding theorem

If  $H(X) \leq C$  then it is possible to transmit through a memoryless channel with capacity  $C$  an amount of information  $H(X)$  with arbitrarily small probability of error.

Else, it is possible to encode the source in such a way that  $H(X/Y) < H(X) - C + \epsilon$ , where  $\epsilon$  is arbitrarily small.

There is no coding method which gives  $H(X/Y) < H(X) - C$ .

---

### Cascading of channels

Suppose that  $X$  is the input of the first channel, while its output  $Y$  is again input for the second channel while the output of the second channel is given by  $Z$ . In this case, an amount of information  $H(X)$  is presented at the input and  $H(Z)$  is received at the output, while  $H(Y)$  passes between the two.

For such case, when a channel with transmission rate  $R_0$  is followed by a second channel with transmission rate  $R$  then we have that:

$$R \leq R_0$$

The theorem above is known as the *data processing theorem*. In actual fact the supposition indicates that only a loss of information is possible when processing data successively.

---

### The continuous communication channel

#### Capacity of a non-distorted continuous information channel

For a non-distorted ( $H(X/Y) = H(Y/X) = 0$ ) continuous information channel we have that  $\max_{p(x)}$  of the received signal is more complicated due to the fact that extra restrictions usually need to be imposed on the channel. These restrictions may for example be a bounded amplitude or a constant variance.

For the case of a bounded amplitude in a range of  $-A$  to  $+A$ , we have  $\max_{p(x)} = \frac{1}{2A}$  (p follows an uniform distribution), for the case of a constant variance, we have that



$\max_{p(x)} = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$  (p follows the normal distribution), hence if such channel receives  $m$  different symbols we have:

Bounded amplitude:  $C = \max_{p(x)} I(X;Y) = H(X) = \log(2A)$ . [bits/symbol]

Constant variance:  $C = \max_{p(x)} I(X;Y) = H(X) = \log(\sigma\sqrt{2\pi e})$ . [bits/symbol]

---

## Appendix

### Non-optimal code word length

The size  $L$  of a not optimized code word length out of a code alphabet  $A$  containing  $q$  symbols that can fully encode a source  $S$  containing  $n$  symbols is  $L = \log_q n$

---

### Optimal encoding

**If a code is an optimal code**, then **the length of the code word** should be exactly **equal** (or least upper integer bound) to the **entropy of the source**.

---

### Error probability of a transmission

In a binary system, the error probability of the transmission is:

$$p_{err} = 2^{N(R-C)}$$

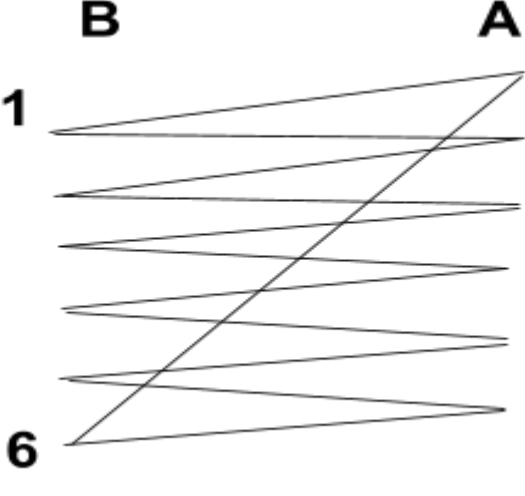
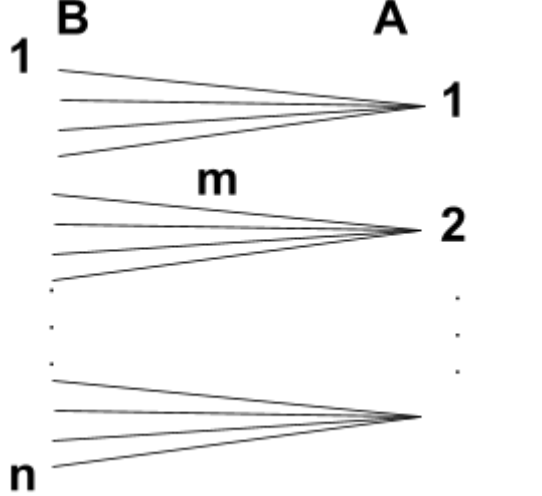

---

### Amount of typical sets of a discrete stochastic variable

$$n = 2^{H(X)}$$

## Calculating channel capacity of trivial examples

$$C = \text{Max}_{p(b)} I(A, B) = S(B) - S(B/A)$$

	
$C = \log_2(6) - \log_2(2) = \log_2(3)$	$C = \log_2(n) - \log_2(m) = \log_2(\frac{n}{m})$

Given information is measured in bits,  $C \cdot 2$  is the maximum number of entries that can be used by the channel so that  $\mathbf{A} \rightarrow \mathbf{B}$  can become bijective.

## References

- [Information Theory - Jan C A van der Lubbe](#)
- [Information Theory, a short course by a physicist - Gregory Falkovich](#)