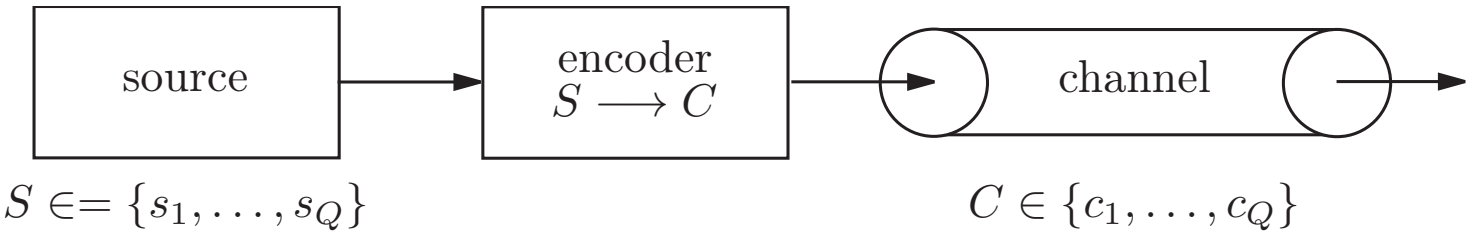
Information Theory and Coding

Discrete source coding

Cédric RICHARD
Université Côte d'Azur

Discrete source coding

Each of the Q states s_i of source S is associated with a codeword, that is, a sequence of n_i symbols of a q-ary alphabet. These constitute a source code that can be noted as follows: $\mathcal{C} = \{c_1, \ldots, c_q\}$.



Example. The Morse code

- ▷ quaternary code (dot, dash, long space, short space)
- ▷ variable length code
- ▷ the shortest sequence is for "E"

Problem

Source coding and adaptation (ideal noiseless channel)

Let S be a source characterized by a rate D_s (Q-ary symbol per second). Consider a noiseless channel with maximum rate D_c (q-ary symbol per second). We define:

- emission rate of the source : $T \triangleq D_s\, H(S)$
- channel capacity : $C \triangleq D_c \log q$

If T > C: the channel cannot transmit the information

If T $\leq$ C : the channel can theoretically transmit the information

If we have a q-ary code where the average length $\overline{n}$ of codewords is such that $\overline{n}\, D_s \leq D_c$, then this code can be used for transmission.

Otherwise, how to encode the source words to make their transmission possible?

Source coding is used to eliminate redundant information WITHOUT LOSS !!!

Discrete-time source

General model

A discrete source S is defined by an alphabet $\mathcal{A} = \{s_1, \ldots, s_Q\}$ and an emission mechanism. It is a discrete-time random process

$$S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots$$

characterized by joint laws:

$$P(S_1, \ldots, S_n), \forall n \in \mathbb{N}^*$$

▷ **model too general to give rise to tractable developments**

Discrete-time source

Complementary assumptions

For simplicity, assumptions need to be made about the source.

Property 1 (Stationary process)**.** *A random process S_i is said to be stationary if the laws that govern it are independent of the origin of time, that is,*

$$P(S_1 = s_{i_1}, \ldots, S_n = s_{i_n}) = P(S_{n_0+1} = s_{i_1}, \ldots, S_{n_0+n} = s_{i_n}),$$

for all positive n_0 and n.

Example. A memoryless source is characterized by independent and identically distributed S_i. This is a stationary process.

$$P(S_1 = s_{i_1}, \ldots, S_n = s_{i_n}) = P(S = s_{i_1}) \ldots P(S = s_{i_n}).$$

Discrete-time source

Complementary assumptions

Again, for the sake of simplicity, the following ergodicity assumption can be made.

Property 2 (Ergodic process)**.** *A stationary random process S_i is ergodic if, for every $k = 1, 2, \ldots$, for every set of indices $i_1, \ldots, i_k$ and for any bounded function $f(\cdot)$ from $\mathcal{A}^k$ into $\mathbb{R}$, we have:*

$$\frac{1}{n} \sum_{k=1}^{n} f(S_{i_1}, \ldots, S_{i_k}) \xrightarrow{a.s.} E\{f(S_{i_1}, \ldots, S_{i_k})\}.$$

Interest. An ergodic process can be studied by observing any long enough trajectory.

Discrete-time source

Markov source

Any source S emits symbols according to a law that can depend on all past symbols.

Definition 1 (Markov source)**.** *A source S is said to be Markovian if*

$$P(S_{n+1} = s_{i_{n+1}} | S_n = s_{i_n}, \ldots, S_1 = s_{i_1}) = P(S_{n+1} = s_{i_{n+1}} | S_n = s_{i_n})$$

for all $s_{i_1}, \ldots, s_{i_{n+1}}$ in $\mathcal{A}$.

As a direct consequence we have

$$P(S_1, \ldots, S_n) = P(S_1)\, P(S_2|S_1)\, \ldots\, P(S_n|S_{n-1})$$

Discrete-time source

Markov source

Definition 2 (Time invariance)**.** *A Markov source S is time-invariant if, for all $n \in \{1, 2, \ldots\}$, we have*

$$P(S_{n+1}|S_n) = P(S_2|S_1)$$

Such a source is entirely defined by the vector of initial probabilities $p|_{t=0}$ and the transition Π whose entries are

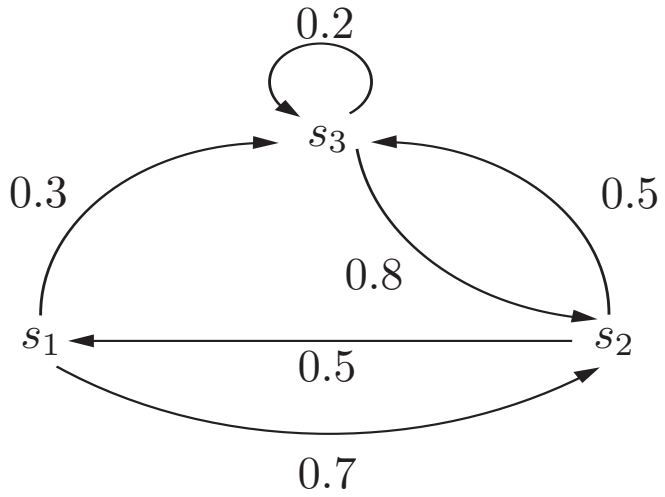
$$\Pi(i, j) = P(S_2 = s_j|S_1 = s_i)$$

Obviously, we have $\sum_{j=1}^{q} \Pi(i, j) = 1$ et $\Pi(i, j) \geq 0$.

Discrete-time source

Example of Markov source

Consider the following Markov source



The corresponding transition matrix can be written as:

$$\Pi = \begin{pmatrix} 0 & 0.7 & 0.3 \\ 0.5 & 0 & 0.5 \\ 0 & 0.8 & 0.2 \end{pmatrix}$$

Discrete-time source

Markov source in steady state

Definition 3 (steady-state - version 1)**.** *Consider a Markov source S. If it exists, its steady state distribution is defined as:*

$$\lim_{n\to\infty} P(S_n = s_i)$$

for all $i \in \{1, \ldots, Q\}$.

Let $p|_{t\to\infty}$ the steady-state distribution if it exists. Given that $p|_{t=n} = p|_{t=n-1}\,\Pi$, we have:

$$p|_{t\to\infty} = p|_{t\to\infty}\,\Pi$$

We say that $p|_{t\to\infty}$ is the steady-state distribution of S since initializing it with $p|_{t\to\infty}$ makes it stationary.

Drawback. The steady state defined in this way depends on the initial distribution $p|_{t=0}$. Other definitions exist.

Discrete-time source

Markov source in steady state

Definition 4 (steady-state - version 2)**.** *Consider a Markov source S. If it exists, its steady state distribution is defined as:*

$$\lim_{n\to\infty} P(S_n = s_i | S_1 = j)$$

for all $i, j \in \{1, \ldots, Q\}$.

Main interest. The asymptotic behavior of S is independent of the initial distribution.

Discrete-time source

m-th order Markov source

A Markov source is characterized by a memory of size $m = 1$. This can be generalized to memory sizes $m > 1$.

Definition 5 (m-th order Markov source)**.** *A source S is an m-th order Markov source if:*

$$P(S_{n+1} = s_{i_{n+1}} | S_n = s_{i_n}, \ldots, S_1 = s_{i_1}) = P(S_{n+1} = s_{i_{n+1}} | S_{n-m} = s_{i_{n-m}}, \ldots, S_n = s_{i_n})$$

for all $s_{i_1}, \ldots, s_{i_{n+1}}$ in $\mathcal{A}$.

Remark. Any m-th order Markov source S can be expressed as a 1-st order Markov source by considering an m-th order extension of S.

Discrete-time source

Entropy of a stationary source

Any source S emits symbols according to a law that can depend on the symbols that came before them. The definition of the entropy of S must take this into account.

Definition 6 (Entropy of a stationary source - version 1)**.** *The entropy of a stationary S source is defined as:*

$$H_0 \triangleq \lim_{n \to +\infty} H(S_n | S_1, \ldots, S_{n-1}).$$

This definition only makes sense if the limit exists.

Discrete-time source

Entropy of a stationary source

Validation of the definition. One need to check that the following limit exists:

$$\lim_{n \to +\infty} H(S_n | S_1, \ldots, S_{n-1})$$

We have:

$$0 \leq H(S_n | S_1, S_2, \ldots, S_{n-1}) \leq H(S_n | S_2, \ldots, S_{n-1}) \leq \ldots \leq H(S_n).$$

Since S is stationary, we can write:

$$H(S_n) = H(S_1) \qquad H(S_n | S_{n-1}) = H(S_2 | S_1) \qquad \ldots$$

The above inequality can be replaced by:

$$0 \leq H(S_n | S_1, \ldots, S_{n-1}) \leq H(S_{n-1} | S_1, \ldots, S_{n-2}) \leq \ldots \leq H(S_1).$$

The series $\{H(S_n | S_1, \ldots, S_{n-1})\}_{n \geq 1}$ is decreasing and bounded. It is therefore convergent, ensuring the validity of the definition in the stationary case.

Discrete-time source

Entropy of a stationary source

Definition 7 (Entropy of a stationary source - version 2)**.** *The entropy of a stationary S source is defined as:*

$$H_0 \triangleq \lim_{n \to +\infty} \frac{H(S_1, \ldots, S_n)}{n}.$$

Both definitions are equivalent in the case of stationary sources. Indeed, it results from the following equality:

$$H(S_1, \ldots, S_n) = H(S_1) + H(S_2|S_1) + \ldots + H(S_n|S_1, \ldots, S_{n-1})$$

that $H(S_1, \ldots, S_n)/n$ is the arithmetic mean of the n first terms of the series $H(S_1)$, $H(S_2|S_1)$, $\ldots$, $H(S_n|S_1, \ldots, S_{n-1})$. Cesaro's theorem yields the expected result.

Cesaro's theorem. If $a_n \underset{n \to \infty}{\longrightarrow} a$, then $\frac{1}{n} \sum_{k=1}^{n} a_k \underset{n \to \infty}{\longrightarrow} a$

Discrete-time source

Entropy of a stationary source

Example 1. In the case of a memoryless source, characterized by independent and identically distributed S_i, we have:

$$H_0 = H(S_1).$$

Example 2. If S denotes a time-invariant Markov source, its entropy is given by:

$$H_0 = H(S_2|S_1).$$

Source coding

Definitions

Source coding consists of associating to each symbol s_i generated by a source, a sequence of symbols of a q-ary alphabet, referred to as a codeword.

Example 1. ASCII (7 bits) et extended ASCII (8 bits), Morse code, etc.

Example 2.

	code A	code B	code C	code D	code E	code F	code G
s_1	1	0	00	0	0	0	0
s_2	1	10	11	10	01	10	10
s_3	0	01	10	11	011	110	110
s_4	0	11	01	110	0111	1110	111

SOURCE CODING

Definition

Regularity. A code is said to be nonsingular if all codewords are distinct.

Decodability.

A nonsingular code is called uniquely decodable if any sequence of codewords can be decoded only in a unique way.

> **Fixed length.** With fixed-length codewords, any message can be decoded without ambiguity.
>
> **Separator.** A symbol of the alphabet is used as a word separator.
>
> **Without prefix.** A code is called a prefix code or an instantaneous code if no codeword is a prefix of any other codeword.

Exercise. Characterize codes A to G.

Toward Shannon's first theorem

Kraft's inequality

We propose to design uniquely decodable codes, and more particularly instantaneous codes, that are as compact as possible.

Kraft's inequality provides a necessary and sufficient condition on the existence of instantaneous codes for given codeword lengths.

Theorem 1 (Kraft's inequality)**.** *Let $n_1, \ldots, n_Q$ be candidate codeword lengths to encode the Q symbols of a source with a q-ary alphabet. A necessary and sufficient condition for the existence of an instantaneous code with these codeword lengths is given by:*

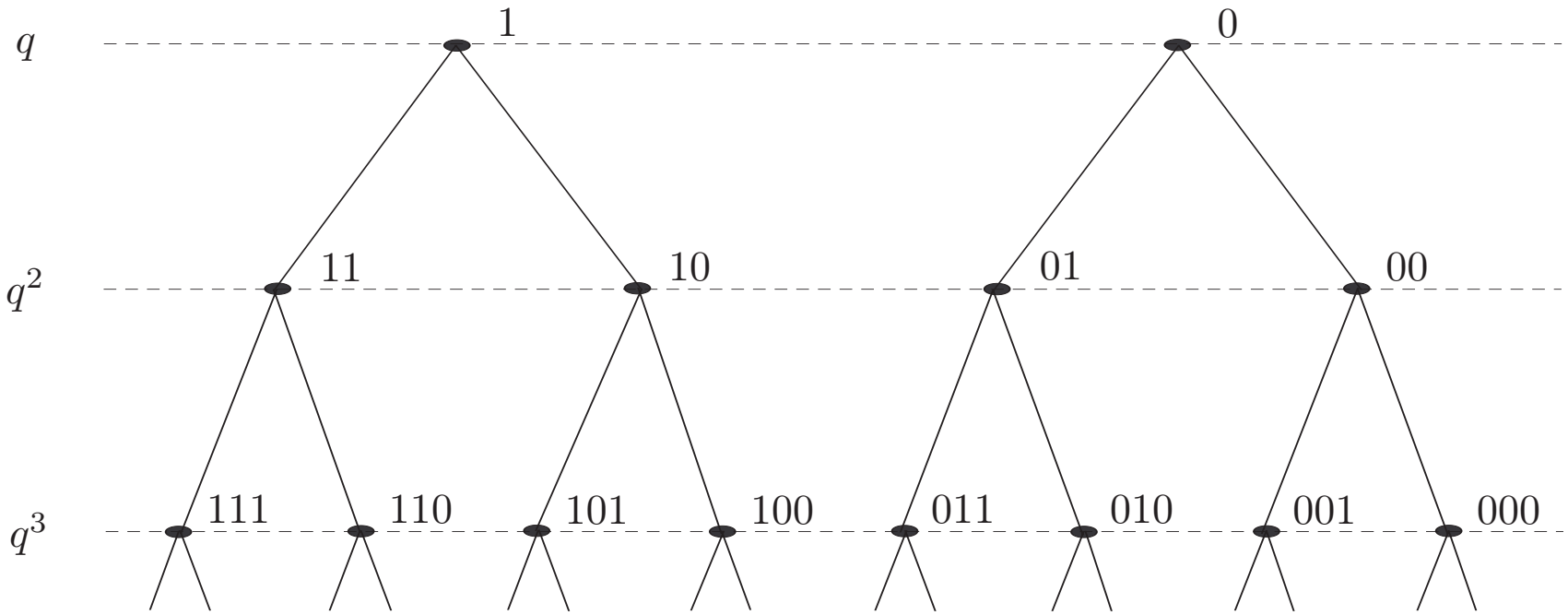
$$\sum_{i=1}^{Q} q^{-n_i} \leq 1.$$

Remark. The same necessary and sufficient condition was established by McMillan for uniquely decodable codes, previously to Kraft's inequality.

Toward Shannon's first theorem

Kraft's inequality

Proof. The following representation, in the case of a binary code, makes the proof clear.



Toward Shannon's first theorem

Kraft's inequality

Let $n_1 \leq \ldots \leq n_Q$. Consider a q-ary tree of height n_Q. This tree then has q^{n_Q} leaf nodes.

Necessary condition. The prefix condition requires that a codeword of length n_i excludes $q^{n_Q - n_i}$ leaf nodes. Therefore, the total number of excluded leaf nodes must satisfy:

$$\sum_{i=1}^{Q} q^{n_Q - n_i} \leq q^{n_Q}.$$

Sufficient condition.

First, we select a node with depth n_1, which excludes $q^{n_Q - n_1}$ leaf nodes. However, there are still available leaf nodes because, using Kraft's inequality, we know that

$$q^{n_Q - n_1} < q^{n_Q}$$

On the way to one of the available leaf nodes, we select a node with depth n_2, ...

Toward Shannon's first theorem

McMillan's inequality

Kraft's inequality implies that McMillan's inequality is sufficient since any prefix code is uniquely decodable.

Necessary condition. Consider the following expansion:

$$\left(\sum_{i=1}^{Q} r_i\, q^{-i}\right)^N = \sum_{n=1}^{NQ} \nu(n)\, q^{-n}$$

where $\nu(n) = \sum_{i_1+\ldots i_N=n} r_{i_1} \ldots r_{i_N}$. Interpreting r_i as the number of codewords of length i, $\nu(n)$ corresponds to the number of messages of length n. The unique decodability condition implies that $\nu(n) \leq q^n$. Then we have:

$$\sum_{i=1}^{Q} r_i\, q^{-i} \leq (NQ)^{\frac{1}{N}},$$

which leads to the result by considering the limit of the upper bound when N tends to infinity.

Toward Shannon's first theorem

McMillan's inequality

Definition 8 (Complete code)**.** *A code is complete if:*

$$\sum_{i=1}^{Q} q^{-n_i} = 1.$$

Toward Shannon's first theorem

McMillan's inequality

As an example, McMillan's inequality is applied to different codes.

	code A	code B	code C
s_1	00	0	0
s_2	01	100	10
s_3	10	110	110
s_4	11	111	11
$\sum_{i=1}^{4} 2^{-n_i}$	1	7/8	9/8

Codes A and B are uniquely decodable, the first one being complete. Code C is not uniquely decodable.

Toward Shannon's first theorem

Consequences of McMillan's inequality

Let S be a memoryless source with Q symbols. Let p_i be the probability of symbol s_i, encoded to a q-ary codeword of length n_i. By setting:

$$q_i = \frac{q^{-n_i}}{\sum_{j=1}^{Q} q^{-n_j}},$$

then applying Gibb's inequality with p_i and q_i, we obtain:

$$\sum_{i=1}^{Q} p_i \log \frac{1}{p_i} + \sum_{i=1}^{Q} p_i \log q^{-n_i} \leq \log \sum_{i=1}^{Q} q^{-n_i}.$$

Applying McMillan's inequality to the right-hand side member of the inequality yields:

$$H(S) - \overline{n} \log q \leq \log \sum_{i=1}^{Q} q^{-n_i} \leq 0,$$

where $\overline{n} = \sum_{i=1}^{Q} p_i \, n_i$ is the expected length of the codewords.

Toward Shannon's first theorem

Consequences of McMillan's inequality

Theorem 2. *The expected length $\overline{n}$ of the codewords of any uniquely decodable code is lower-bounded by:*

$$\frac{H(S)}{\log q} \leq \overline{n}.$$

Condition of equality. The above inequality turns into an equality if $\sum_{i=1}^{Q} q^{-n_i} = 1$, that is, $p_i = q^{-n_i}$. This means that:

$$n_i = \frac{\log \frac{1}{p_i}}{\log q}.$$

Definition 9. *Any code where each codeword i is of length $n_i = \frac{\log \frac{1}{p_i}}{\log q}$ is absolutely optimal.*

Toward Shannon's first theorem

Consequences of McMillan's inequality

Usually, the above equality condition is not satisfied because $n_i = \frac{\log \frac{1}{p_i}}{\log q}$ is not an integer. However, it is possible to construct a code such that:

$$\frac{\log \frac{1}{p_i}}{\log q} \leq n_i < \frac{\log \frac{1}{p_i}}{\log q} + 1.$$

Multiplying each member by p_i and summing over i, we obtain:

$$\frac{H(S)}{\log q} \leq \overline{n} < \frac{H(S)}{\log q} + 1.$$

Definition 10 (Shannon's code: predefined codeword lengths)**.** *We talk about a Shannon's code when:*

$$n_i = \left\lceil \frac{\log \frac{1}{p_i}}{\log q} \right\rceil.$$

Shannon's first theorem

Statement and demonstration

The bounds that have just been established will allow us to demonstrate Shannon's first theorem, which reads as follows:

Theorem 3. *For any stationary source, there is a coding process to design a uniquely decodable code where the expected codeword length is as close to its lower bound as you want it to be.*

Proof in the case of a memoryless source. Consider the k^{th} extension of source S. In the case of a memoryless source:

$$\frac{kH(S)}{\log q} \leq \overline{n}_k < \frac{kH(S)}{\log q} + 1.$$

In this expression, $\overline{n}_k$ denotes the expected length of the codewords used to encode the k^{th} extension of source S. Dividing by k and calculating the limit as k tends to infinity leads to the result.

Premier théorème de Shannon

énoncé et démonstration

Proof in the case of a stationary source. Consider the k^{th} extension of a source S. In the case of a memoryless source, we have:

$$\frac{H(S_1, \ldots, S_k)}{k \log q} \leq \frac{\overline{n}_k}{k} < \frac{H(S_1, \ldots, S_k)}{k \log q} + \frac{1}{k}.$$

In this expression, $\overline{n}_k$ denotes la longueur denotes the expected length of the codewords used to encode the k^{th} extension of source S.

In the case of a stationary source, we know that $\lim_{k \to \infty} H(S_1, \ldots, S_k)$ exists. Denoting this limit by H_0 yields:

$$\lim_{k \to \infty} \frac{\overline{n}_k}{k} = \frac{H_0}{\log q}.$$

Binary coding techniques

Shannon's code: predefined codeword length

Shannon's first theorem provides an asymptotic property, but do not provide any practical method for doing so.

Shannon's coding technique consists of associating n_i q-ary symbols to each source state s_i, where:

$$n_i = \left\lceil \frac{\log \frac{1}{p_i}}{\log q} \right\rceil .$$

Binary coding techniques

Shannon's code: predefined codeword length

We consider a 5-symbol source $\{s_1, \ldots, s_5\}$ defined by probabilities:

$$\begin{array}{lll} p_1 = 0.35 & -\log_2 p_1 = 1.51 & \longrightarrow \quad n_1 = 2 \\ p_2 = 0.22 & -\log_2 p_2 = 2.18 & \longrightarrow \quad n_2 = 3 \\ p_3 = 0.18 & -\log_2 p_3 = 2.47 & \longrightarrow \quad n_3 = 3 \\ p_4 = 0.15 & -\log_2 p_4 = 2.73 & \longrightarrow \quad n_4 = 3 \\ p_5 = 0.10 & -\log_2 p_5 = 3.32 & \longrightarrow \quad n_5 = 4. \end{array}$$

We can easily get an instantaneous code that satisfies the above conditions on n_i using a tree. For instance:

$$s_1 : 00 \qquad s_2 : 010 \qquad s_3 : 011 \qquad s_4 : 100 \qquad s_5 : 1010.$$

This leads to $\overline{n} = 2.75$, to be compared to $H(S) = 2.19$ Sh/symb.

Binary coding techniques

Shannon-Fano's code

Shannon-Fano's code is the first code that started to exploite the redundancy of a source. Its principle is now outlined.

1. Arrange the states of the system by decreasing probabilities.
2. Split the system states into 2 groups G_0 et G_1 with probabilities as close as possible without *modifying* their arrangement in 1.
3. Each group G_i is split into 2 sub-groups G_{i0} et G_{i1} with probabilities as close as possible to each other, again without modifying the state arrangement.
4. The procedure stops when each subgroup consists of a single element. The index of the group gives the codeword.

Binary coding techniques

Shannon-Fano's code

To design a Shannon-Fano's code, we proceed as follows:

state	p_i	step 1	step 2	step 3	code
s_1	0.35	0	0		00
s_2	0.22	0	1		01
s_3	0.18	1	0		10
s_4	0.15	1	1	0	110
s_5	0.10	1	1	1	111

This leads to $\overline{n} = 2.25$, to be compared to $H(S) = 2.19$ Sh/symb.

Binary coding techniques

Huffman's code

Huffman's method provides a compact instantaneous code of minimum average length. To achieve this, it exploits the following property.

Lemme 1. *For any source, there is an instantaneous code of minimum expected length that satisfies the following properties.*

1. *If* $P(S = s_i) > P(S = s_j)$*, then* $n_i \leq n_j$*.*
2. *The two longest words, therefore associated with the least likely states, have the same length and differ by only one bit.*

Huffman's method involves grouping the two least likely states together and then treating them as one by summing their probabilities. This technique is then repeated on the remaining states until only two remain.

Binary coding techniques

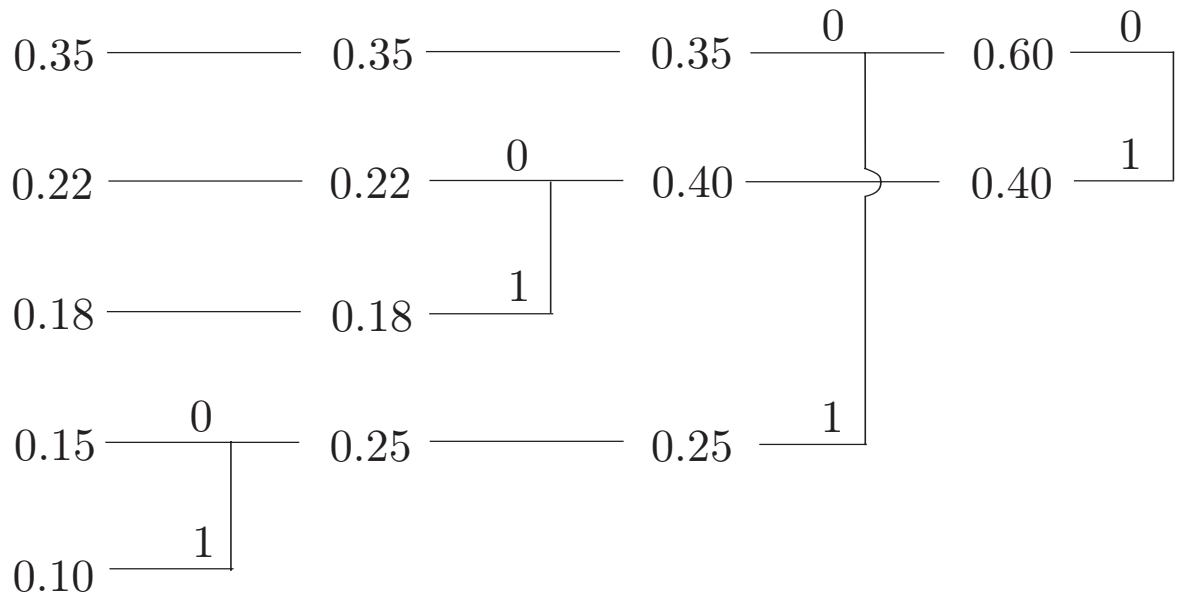
Huffman's code

A tree is built from the leaf nodes, which represent the states of the source.

1. At each step, the two least likely leaves are merged into one.
2. The procedure stops when the result is a single leaf consisting of all the symbols.
3. The reverse path of the tree provides the code words.

Binary coding techniques

Huffman's code



The reverse exploration of the tree provides the following code words:

$$s_1 : 00 \qquad s_2 : 10 \qquad s_3 : 11 \qquad s_4 : 010 \qquad s_5 : 011.$$

This leads to $\overline{n} = 2.25$, to be compared to $H(S) = 2.19$ Sh/symb.