Fundamentals of Mathematical Informatics

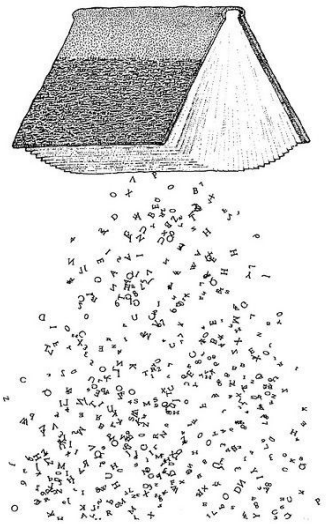
The Noiseless Coding Theorem for Information Sources

Francesco Buscemi

Lecture Two

Information sources

Letters arefalling...



https://www.pinterest.jp/pin/602145412649166203/

Let $W = \{w_1, w_2, \cdots, w_N\}$ be some finite set, e.g., the English alphabet $\{$a, b, $\cdots$, x, y, z$\}$.

Imagine a device that, at each use, emits one element of W drawn at random with probability $\Pr\{\text{`output is } w_j\text{'}\} = p_j$.

Each use of such a device is modeled by a RV W, with range equal to W and $\Pr\{W = w_j\} = p_j$.

Imagine now that such a device can be reused an arbitrary number of times: then we have an **i.i.d. information source** S , namely, a sequence $(W_1, W_2, \cdots, W_i, \cdots)$ of independent and identically distributed (i.i.d.) RVs W_i, all with the same range W and the same probability distribution $\Pr\{W_i = w_j\} = p_j$.

The **entropy rate** of an i.i.d. information source S is $H(S) \stackrel{\text{def}}{=} H(p_1, \cdots, p_N)$.

Question. Take W to be the English alphabet. What is the probability of emitting the particular sequence (h,e,l,l,o,w,o,r,l,d)?

Answer. $\Pr\{(\text{h,e,l,l,o,w,o,r,l,d})\} = \Pr\{\text{h}\} \times \Pr\{\text{e}\} \times \Pr\{\text{l}\} \times \Pr\{\text{l}\} \times \Pr\{\text{o}\} \cdots \times \Pr\{\text{d}\}$.

Information sources can be 'compressed'

Imagine an i.i.d. information source that, at each use, emits one symbol chosen among eight possible ones {a,b,c,d,e,f,g,h} with probability distribution $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64})$.

Imagine now that we want to communicate the source output via a digital channel.

Uniform binary encoding: three bits per symbol, i.e., 'a' *1*→ 000, 'b' *1*→ 001, 'c' *1*→ 010, 'd' *1*→ 011, 'e' *1*→ 100, 'f' *1*→ 101, 'g' *1*→ 110, 'h' *1*→ 111.

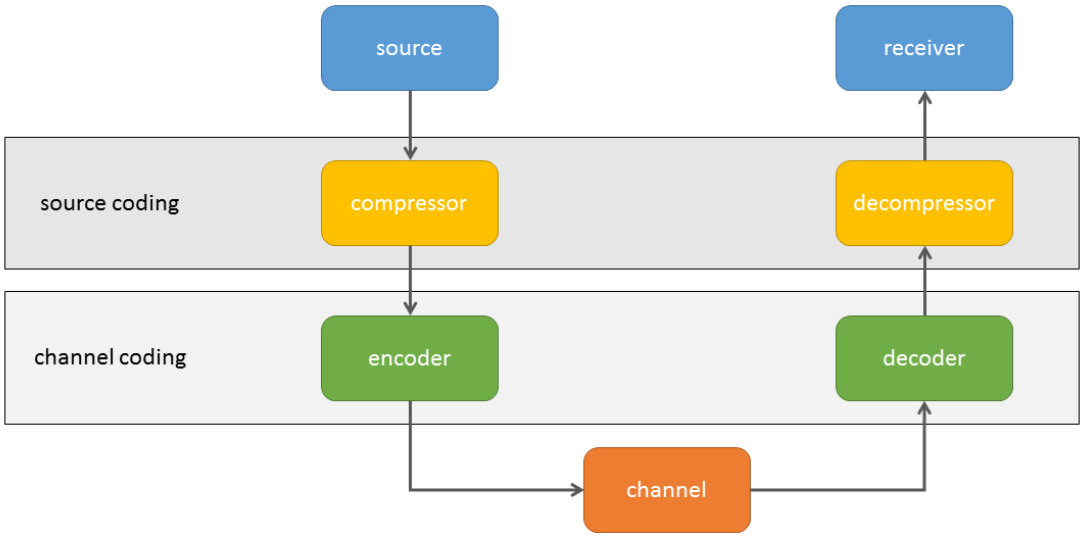
But not all letters happen with the same probability! Can we exploit this?

Better encoding: 'a' *1*→ 0, 'b' *1*→ 10, 'c' *1*→ 110, 'd' *1*→ 1110, 'e' *1*→ 111100, 'f' *1*→ 111101, 'g' *1*→ 111110, 'h' *1*→ 111111.

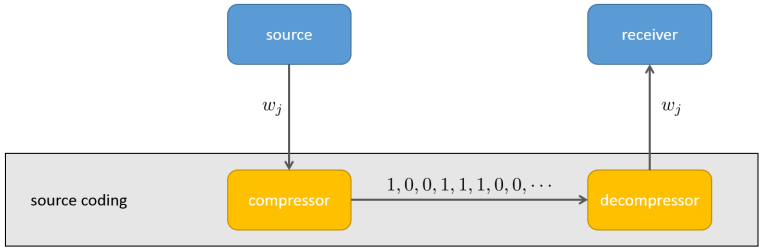
In average, we need to send only 2 bits per use of the source!

Remark. The entropy of this source (remember the horse race?) is also equal to 2 bits... Is this a coincidence or not? Can we do better?

General communication scheme



Encoding (compressor)



Take an i.i.d. information source S whose single use is modeled by a RV W with range $W = \{w_1, \dots, w_N\}$ and probability distribution $\Pr\{W = w_j\} = p_j$. The elements of W are called *source words*. A finite string of source words is called a *message*. The set of all possible messages is denoted by W^*.

Example. In the binary case $W = \{0, 1\}$,
$W^* = \{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, \cdots\}$.

Consider now a D-ary alphabet $\Sigma = \{\sigma_1, \dots, \sigma_D\}$, and the set Σ^* of all finite strings of elements of Σ.

A D-ary *encoding* or *code* is a map $f : W \to \Sigma^*$. The strings $f(w_i)$ are called the *codewords*, and the integers $l_i \stackrel{\text{def}}{=} |f(w_i)|$ are called the *word lengths*.

The **average length** or **rate** of a code is defined as: $\ell(f) \stackrel{\text{def}}{=} \sum_{i=1}^{N} p_i l_i$.

Noiseless encoding: uniquely decodable codes and prefix codes

Given a code $f : W \to \Sigma^*$, messages in W^* are encoded word by word. This defines the *extended code* $f^* : W^* \to \Sigma^*$.
Example. Take $W = \{w_1, w_2, w_3\}$. Given the four-word message $m = w_1w_2w_3w_2 \in W^*$ the corresponding extended code is $f^*(m) = f(w_1)f(w_2)f(w_3)f(w_2) \in \Sigma^*$.
If f^* is injective, the code f is called *uniquely decodable* or *uniquely decipherable*.
A code f is called *instantaneous* or *prefix code* if there do not exist distinct words w_i and w_j such that $f(w_i)$ is a prefix of $f(w_j)$.

Example. Take $W = \{\text{e,r,s,t}\}$, $\Sigma = \{0, 1\}$, and the encoding $f : W \to \Sigma^*$ given by

$$f(\text{e}) = 0, \quad f(\text{r}) = 10, \quad f(\text{s}) = 110, \quad f(\text{t}) = 1110.$$

Try to decode '11101000110'. ('trees')
The above is an example of an instantaneous code.
Instantaneous codes are all uniquely decodable. But not viceversa.

Example. Take again $W = \{\text{e,r,s,t}\}$, $\Sigma = \{0, 1\}$, and the encoding $g : W \to \Sigma^*$ given by

$$g(\text{e}) = 0, \quad g(\text{r}) = 01, \quad g(\text{s}) = 01011, \quad g(\text{t}) = 01011011.$$

Try to decode '01001011001011011'. ('reset')
Prefix codes are 'better' because they can be decoded on line (i.e., without having to wait until the end of the message).

Existence of prefix codes 1/2

Theorem (Kraft's Inequality). Consider a source with $W = \{w_1, \cdots, w_N\}$, and a D-ary alphabet $\Sigma = \{\sigma_1, \cdots, \sigma_D\}$. Then, there exists a prefix code $f : W \to \Sigma^*$ with word lengths $l_1, l_2, \cdots, l_N$ iff $\sum_{i=1}^{N} D^{-l_i} \leq 1$.

Proof of the 'if' part. Assume that the l_i's satisfy Kraft's inequality. Rewrite it as $\sum_{j=1}^{L} n_j D^{-j} \leq 1$, where $L \stackrel{\text{def}}{=} \max_i l_i$ and n_j is the number of l_i's equal to j.

Since the elements of the sum are all positive, $\sum_{j=1}^{L} n_j D^{-j} \leq 1 \Rightarrow \sum_{j=1}^{L-1} n_j D^{-j} \leq 1 \Rightarrow \cdots \Rightarrow n_2 D^{-2} + n_1 D^{-1} \leq 1 \Rightarrow n_1 D^{-1} \leq 1$.

The above inequalities suggest how to construct a prefix code with given word lengths.

Since $n_1 \leq D$, we can use the first n_1 symbols in Σ as codewords of length 1.

There are $D - n_1$ symbols left unused in Σ. We can then form $(D - n_1)D$ words of length 2 writing another letter to their right.

Since $n_2 \leq (D - n_1)D$, we can choose n_2 of them to become the next codewords.

There are now $D^2 - n_1 D - n_2$ length 2 words left unused. Adding one new symbol at their right, we obtain $D^3 - n_1 D^2 - n_2 D$ length 3 words.

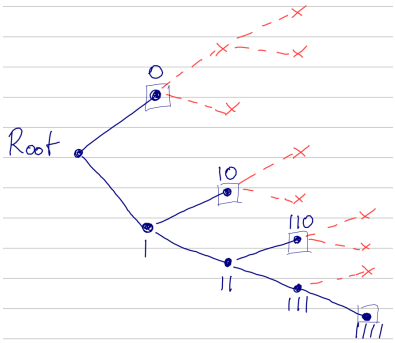
Since $n_3 \leq D^3 - n_1 D^2 - n_2 D$, we can choose...

...

It is clear then that Kraft's inequality, if obeyed, guarantees that a prefix code of given word lengths can be constructed. D

Existence of prefix codes 1/2

Proof of the 'only if' part. Consider a D-ary tree (each node has D children).
Example: the binary tree.



Attach an element of Σ^* to each node as in the figure above. The prefix condition requires that any node corresponding to a codeword ends there.

Let L denote the maximum word length in the code (in the picture, $L = 4$). Consider all the nodes of the tree at level L. A codeword at level l_i has D^{L-l_i} descendants at level L. Each of these descendants sets must be disjoint, and the total number of nodes in these sets must be less than or equal to D^L. Hence, summing up over all codewords, we have

$$\sum_{i=1}^{N} D^{L-l_i} \leq D^L,$$

that is $\sum_{i=1}^{N} D^{-l_i} \leq 1$. D

The noiseless source coding theorem

Let S be an i.i.d. information source of words $w_1, \cdots, w_N$ with probabilities $p_1, \cdots, p_N$, respectively, and entropy rate $H(S) \stackrel{\text{def}}{=} H(p_1, \cdots, p_N)$. Take also a D-ary alphabet $\Sigma = \{\sigma_1, \cdots, \sigma_D\}$.

Theorem. Any D-ary prefix code $f: W \to \Sigma^*$ must satisfy the following inequality:

$$\ell(f) \stackrel{\text{def}}{=} \sum_{i=1}^{N} p_i l_i \geq \frac{H(S)}{\log_2 D}.$$

Moreover, there always exists a D-ary prefix code $\bar{f}: W \to \Sigma^*$ such that

$$\ell(\bar{f}) < \frac{H(S)}{\log_2 D} + 1.$$

Conclusion. 'Good' D-ary prefix codes are those with rate bounded as $\frac{H(S)}{\log_2 D} \leq \ell(f) < \frac{H(S)}{\log_2 D} + 1$.

Proof of the lower bound (converse part)

Let $f: W \to \Sigma^*$ be a D-ary prefix code for S. Then, it must obey Kraft's inequality, i.e., $\sum_{i=1}^{N} D^{-l_i} \leq 1$.

For $G \stackrel{\text{def}}{=} \sum_{i=1}^{N} D^{-l_i}$, define the probability distribution $q_i \stackrel{\text{def}}{=} D^{-l_i}/G$.

By the Key Lemma,

$$H(S) \stackrel{\text{def}}{=} H(p_1, \cdots, p_N) = -\sum_{i=1}^{N} p_i \log_2 p_i \leq -\sum_{i=1}^{N} p_i \log_2 q_i.$$

But, by definition, $\log_2 q_i = -\log_2 G - l_i \log_2 D$.

Then, $H(S) \leq \sum_i p_i(\log_2 G + l_i \log_2 D) = \log_2 G + \ell(f) \log_2 D$.

Since $G \geq 1$, $\log_2 G \geq 0$ and, therefore, $H(S) \leq \ell(f) \log_2 D$, i.e.,

$$\ell(f) \geq \frac{H(S)}{\log_2 D}.$$

$\log_2 D$

Proof of the upper bound (achievability)

Imagine that, for each p_i, there exists an integer $\bar{l}_i$ such that $D^{-\bar{l}_i} = p_i$, i.e., $\bar{l}_i = -\frac{\log_2 p_i}{\log_2 D}$.

Then, Kraft's inequality would be automatically satisfied, because $\sum_i D^{-\bar{l}_i} = \sum_i p_i = 1$.

Then, we would know that there exists (and we would know how to construct) a D-ary prefix code $\bar{f}$ with word lengths $\bar{l}_i$.

Its average length would be $L(\bar{f}) = \sum_i p_i \bar{l}_i = \sum_i p_i(-\frac{\log_2 p_i}{\log_2 D}) = \frac{H(S)}{\log_2 D}$.

That would be optimal! We cannot go lower than that!

The only problem is that, in general, the numbers $\bar{l}_i = -\frac{\log_2 p_i}{\log_2 D}$ are not *integer numbers*, and hence are not valid word lengths!

To avoid such a problem, choose $l_i^* = \lceil \bar{l}_i \rceil$ for all i. (The symbol $\lceil x \rceil$ denotes the 'ceiling' of x, i.e., the smallest integer greater than or equal to x.)

This implies that $\bar{l}_i \le l_i^* < \bar{l}_i + 1$ for all i.

Again, Kraft's inequality is obeyed since $\sum_i D^{-l_i^*} \le \sum_i D^{-\bar{l}_i} = 1$.

Only the average length is worse, because $\sum_i p_i l_i^* \ge \sum_i p_i \bar{l}_i$, but not too much, because

$$\sum_i p_i l_i^* < \sum_i p_i(\bar{l}_i + 1) = \frac{H(S)}{\log_2 D} + \sum_i p_i = \frac{H(S)}{\log_2 D} + 1. \quad \text{D}$$

Can we do better with uniquely decodable codes?

All prefix codes are uniquely decodable but not viceversa.

So, if we are happy with uniquely decodable codes (possibly not prefix codes), can we perhaps achieve better rates of compression?

The answer is no.

Theorem (McMillan-Kraft). A prefix code with word lengths $l_1, l_2, \cdots, l_N$ exists iff a uniquely decodable code with the same word lenghts exists.

Block-coding to get sharper bounds

We proved that good D-ary codes are those with $\ell \in \frac{H(S)}{\log_2 D}, \frac{H(S)}{\log_2 D} + 1$

The one-letter overhead is due to the fact that word lengths need to be integer numbers.

However, a technique called **block-coding** allows us to spread the overhead over many source words at once, so that the overhead per source word goes to zero.

Take an i.i.d. information source S of words in $W = \{w_1, \cdots, w_N\}$ with probabilities $p_1, \cdots, p_N$, and group words two by two: we obtain the source S^2, with words in $W^2 = \{w_1w_1, w_1w_2, \cdots, w_1w_N, w_2w_1, w_2w_2, \cdots, w_Nw_N\}$ and probabilities $p_{ij} = p_i p_j$.

Good D-ary codes for S^2 are such that $\ell \in \frac{H(S^2)}{\log_2 D}, \frac{H(S^2)}{\log_2 D} + 1$

But codes for S^2 encode two S-words in each code word! So, for such codes, we consider their *average length per source word*, i.e., $\frac{\ell}{2} \in \frac{1}{2}\frac{H(S^2)}{\log_2 D}, \frac{1}{2}\frac{H(S^2)}{\log_2 D} + \frac{1}{2}$

Since S is i.i.d., $H(S^2) = 2H(S)$. Hence, good codes of S^2 (called **codes of block-length 2**) are such that $\frac{\ell}{2} \in \frac{H(S)}{\log_2 D}, \frac{H(S)}{\log_2 D} + \frac{1}{2}$

For block-length n, $\frac{\ell}{n} \in \frac{H(S)}{\log_2 D}, \frac{H(S)}{\log_2 D} + \frac{1}{n}$

Therefore, for codes of increasingly larger block-length, i.e., for $n \to \infty$, the average length per source word converges to $H(S)$.

Remark. Block-codes can get incredibly complicated as the block-length increases.

Summary of lecture two

In the previous lecture, we argued that the entropy $H(p_1, \cdots, p_N)$ measures 'how uncertain' is a RV.

In this lecture we made this rigorous: the entropy $H(p_1, \cdots, p_N)$ essentially *is* the optimal rate at which an i.i.d. information source outputting words with probabilities $p_1, \cdots, p_N$ can be compressed (using a binary prefix code).

Keywords for lecture two

i.i.d. information source, entropy rate of a source, source words and messages, codes and codewords, average length of a code, uniquely decodable codes, prefix codes, Kraft's inequality, Shannon's noiseless source coding theorem, block-coding