Entropy and information

June 28, 2018

Contents

1 What is information?

Like "work", "heat" "temperature" and "energy" information is a notion that existed before it was given a precise scientific meaning. The scientific definition allows quantifying the concept at the price that it does does not always coincide with the common meanings. In particular, we shall not be able to say "How much information is in the bible" or in Newton's Principia. However, it allows to compress and transmit data without losing information, to encode information in noisy channels and recover from errors.

Information theory quantify how much you learned getting the string x. You may view its as a measure of the "surprise" in the message. The surprise is bigger the larger the panorama of messsages that you expect to receive.

Example 1.1. *Prof. X flunks every student and Prof. Y passes every one. In either case, if you know which professor gave the grade, there is no information in the grade itself. At least not about the student. But, if you do not know which Professor gave the grade then there is information in the grade, if not about the student's knowledge then about the professor.*

2 Kolmogorov Complexity

A conceptually nice definition of the information content of a list x, is the Kolmogorov Complexity, $K(x)$

$$K(x) = \texttt{Length of the shortest algorithm that generates } x \quad (2.1)$$

For example the information given by the string of the first million digits of

$$\pi = 3.1415926\ldots \quad (2.2)$$

can be encoded in a short program, shorter than a million characters, for computing π to this accuracy.

The problems with this definition is that there is no algorithm for computing or even estimating $K(x)$.

Kolmogorov assigns high complexity to a (given) random string. In a truely random string there is, essentially, by definition, no deterministic algorithm to fix it. The best you can do is list the original string. We see that this definition does not attempt to analyze the content of the string x, decide if it is nonsense or profound.

3 Entropy in statistical mechanics

Boltzman defined the entropy as the number of microscopic states available consistent with thermodynamic information, such as total energy (or temperature), volume (or pressure) etc. Inscribed on his grave is his formula

$$S = k_B \ln W \tag{3.1}$$

From now on we shall use unit so that $k_B = 1$.

Consider a large one dimensional system of n "atoms" located at sites $j \in 1, \ldots, N$. Each "atom" can be in m different energy states, labeled by x

$$x \in 1, \ldots, m \tag{3.2}$$

A microscopic configuration tells you the state of each atom

$$x(1) \ldots x(n) \tag{3.3}$$

There are

$$m^N = e^{N \ln m} \tag{3.4}$$

microscopic configurations and the associated entropy with such a system (which corresponds to the case where $T = \infty$) is

$$S = k_B N \ln m = (k_B \ln 2) N \log m \tag{3.5}$$

Where log is in base 2. From now on we shal use units where

$$k_B \ln 2 = 1 \tag{3.6}$$

so entropy is simply

$$S = N \log m \tag{3.7}$$

Now suppose the temperature is finite and I give you the thermodynamic characterization of this system by telling you the fraction of atoms in state s

$$p_E(x) = \frac{\#\text{atoms in energy state x}}{N} \tag{3.8}$$

There is a large number of microscopic configurations compatible with the given $p(x)$, corresponding to permuting the positions of the "atoms".

To compute S in this case is a problem in combinatorics and the answer turns out to be

$$S(E) = -N \sum_{x=1}^{m} p_E(x) \log p_E(x), \tag{3.9}$$

The entropy is extensive as it must be.

This may be taken as a motivation to define a generalization of the notion of entropy to any probability distribution $P_X(x)$, that need not be associated with energy states. In the application to communication

$$P_X(x) \tag{3.10}$$

will normally be the probability distribution for letters in an alpha-bet, and the microscopic configuration is a string of text. The Shannon information associated with P_X is,

$$H(P_X) = -\sum_{x=1}^{m} p_X(x) \log p_X(x), \tag{3.11}$$

and by convention

$$0 \log 0 = \lim_{x \to 0} x \log x = 0 \tag{3.12}$$

Let me derive this result in the simplest case $m = 2$. The number of microscopic configurations corresponding to Np atoms being in $s = 1$ and $(1-p)N$ atoms in $s = 2$ is

$$\binom{N}{pN} = \frac{N!}{(pN)!((1-p)N)!} \tag{3.13}$$

According to Boltzman, the entropy is

$$S = \log \binom{N}{pN} \tag{3.14}$$

Stirling formula says that

$$\log N! = N(\log N - 1) + O(\log N) \tag{3.15}$$

When N is large, it is a good (relative) approximation to keep the terms on the left. Doing so you will find by simple algebra

$$S = -N \underbrace{(p \log p + (1-p) \log(1-p)}_{-H_2(p)} + O(\log N) \tag{3.16}$$

We shall drop the $O(\log N)$ from now on. However,it sometimes useful to remember that this is an approximation, for example, when you think you discovered a logical inconsistency in the theory taken literally, check that the inconsistency is not a consequence of going beyond the approximation.

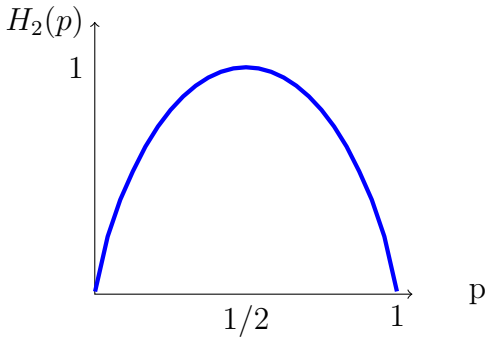


Figure 1: $H_2(p)$ is a concave function of p, symmetric about $p = 1/2$ where $H_2(1/2) = 1$.

4 Shannon entropy and communication theory

e is the most common letter in English with frequency of about 13%. Wikipedia gives the frequencies of all letters. In a large text, you'd expect it to obey these frequencies. We can now play the game we did with the string of atoms with a string of letters and ask for the entropy associated with all strings that satisfy the frequency constraint. A microscopic configuration is then the analog of a string of letters and the Shannon entropy is a measure of how many different text of length N you have. It is, of course, exponentially big

$$O\left(2^{NH}\right) \tag{4.1}$$

You may argue that it is more reasonable to treat all configurations of atoms on equal footing than treat on equal footing all strings of texts. After all most strings will be meaningless. Surprisingly, this turns out to be a fruitful starting point. It turns out that it is best not to try to "understand" messages, but instead focus on their statistical properties.

Indeed, some messages are corrupted and others are encrypted and focusing on the meaning turns out to lead nowhere. Shannon allows for much flexibility in choosing the probabilities P_X and the alphabet according to circumstances.

Example 4.1. *A pool has N lists.*

- *Suppose the probability of receiving any of the lists is the same $p(x) = \frac{1}{N}$. The entropy is*

$$H = -N \times \frac{1}{N} \log \frac{1}{N} = \log N \equiv n$$

Since you can encode the N members of the list by an integer, which you can write with n binary digit the information you got is naturally quantified by n.

You may wonder why is it useful to have freedom in choosing P_X. Here is an example.

Example 4.2. *Alice can encode 4 pieces of data in two bits by choosing probabilities equal*

$$P_A(a,b) = \frac{1}{4}, \quad a,b \in 0,1 \tag{4.2}$$

leading to

$$H(P_A) = \log 4 = 2 \tag{4.3}$$

expressing the fact that each of the four messages

$$00, \quad 01, \quad ,10, \quad 11 \tag{4.4}$$

are equally likely.

Now suppose Alice is worries about the possibility of errors in data transmission and to safeguard against error encode the logical bit by multiple bits, e.g.

$$00 \mapsto \mathbf{0}, \quad 11 \mapsto \mathbf{1} \tag{4.5}$$

Alice sends the logical $\mathbf{0}$ *or* $\mathbf{1}$ *with equal probabilities*

$$P_B(\mathbf{a}) = 1/2, \quad \mathbf{a} \in \mathbf{0},\mathbf{1} \tag{4.6}$$

The corresponding Shannon entropy is

$$H(P_B) = \log 2 = 1 \tag{4.7}$$

expressing the fact that she only sends two messages with equal probability..

5 Typical sequences

Consider the list of N coin tosses made with a biased coin, with probability $p \leq 1$ for head and $q = 1 - p \leq 1$ for tail. When N is large, you expect

$$\#\text{ heads} = Np + O(\sqrt{N}) \tag{5.1}$$

Strings of coin tosses with

$$|\#\text{ heads} - Np| \gg O(\sqrt{N}) \tag{5.2}$$

are extremely rare events when N is large. We shall forget about them. These are the typical coin tosses.

We can partition the 2^N string of head and tails into disjoint sets, obtained by the typical N tosses of (infinitely) many different coins, each one with its own bias, p[1]:

$$L(p, L) = \{\text{Typical N coin tosses of a coin with bias } p\ \} \tag{5.3}$$

Clearly

$$\text{all N strings of head and tail} = \cup_p L(p), \quad L(p) \cap \text{L}(p') = 0 \quad p \neq p' \tag{5.4}$$

Although all strings have N elements, some of them are richer than the others in theh sense that they have *many more different element*. This richness is measured by the entropy:

$$\#\text{ distinct element in } L(p, L) = O\left(2^{NH_2(p)}\right)$$

If $H(p) > H(p')$ then

$$L(p, L) \text{ has exponentially more different elements than } L(p', L) \tag{5.5}$$

There is greater richness, i.e. more information, in a list of N bits that comes from an unbiased coin than from any list of the same length that comes from a biased coin.

Shannon (lossless) compression theorem is basically:

The number distinct lists of length N with a given $H(p)$ is the same, i.e. in 1-1 correspondence, with the lists of length N' and $H(p')$ provided

$$NH(P) = N'H(p') \tag{5.6}$$

In particular, the shortest list is the one corresponding the the maximal entropy H.

The theorem does not tell you how to compress a given string. It only guarantees the existence of compression in the case that the entropy of the lists is not already maximal.

One of the most famous and widely used compression algorithm is Lempel-Ziv, which you may know as zip. It was the PhD work of Avraham Lempel under the supervision of Jacob Ziv, both faculty members of the Technion.

[1]Since the mean is defined up to variance, p takes discrete values, e.g. $p_j = j/\sqrt{N}$ and $j \in 1 \ldots, \sqrt{N}$.

6 Basic properties of Shannon entropy

Let x be a random variable taking values in a set X. I shall write

$$P_X(x) \tag{6.1}$$

for the probability of the event $x \in X$ and

$$H(X) = -\sum_x P_X(x) \log P_X(x) \tag{6.2}$$

- The first basic property is

  $$H(X) \geq 0 \tag{6.3}$$

  This follows from the fact that each term in the sum is positive.

- The second basic property is

  $$H(X) \leq \log |X| \tag{6.4}$$

  where $|X|$ is the number of elements in X. The bound is saturated when all events are equally probable

  $$P_X(x) = \frac{1}{|X|} \tag{6.5}$$

- If we have two independent random variables P_X and P_Y then define the joint probability by

  $$P_{XY}(x, y) = P_X(x) P_Y(y) \tag{6.6}$$

  It is a simple exercise to show that in this case the entropy is additive

  $$H(X, Y) = H(X) + H(Y) \tag{6.7}$$

- In the opposite limit when x determines y

  $$P_{XY}(x, y) = P_X(x) \delta_{y, f(x)} \tag{6.8}$$

  we find

  $$\begin{aligned} H(X, Y) &= \sum_{x,y} P_{XY}(x, y) \log P_{XY}(x, y) \\ &= \sum_x P_X \log P_X(x) \\ &= H(X) \end{aligned}$$

- It can be shown that, in general

  $$H(X) \leq H(X, Y) \leq H(X) + H(Y) \tag{6.9}$$

  The left hand side is called monotonicity and the right hand side is called *sub-additivity.*

7 Mutual information and channel capacity

The most important concept in information theory is the mutual information defined by

$$H(X:Y) = H(X) + H(Y) - H(X,Y) \geq 0 \tag{7.1}$$

(Positive by sub-additivity.)

In communication theory X is the random variable in the input and Y is the random variable at the output. Real channels are noisy:

Example 7.1 (Noisy channel:)**.** *Suppose the channel is noisy so that a bit a sent by Alice is transmitted faithfully to Bob with probability $p \geq 1/2$ and flipped with probability $q = 1 - p \leq 1/2$. This situation is described by the conditional probabilities:*

$$P_{BA}(b|a) = p\delta_{ab} + q\delta_{a\neq b}, \quad a, b, \in 0, 1 \tag{7.2}$$

Suppose Alice sends the bits with equal portability

$$P_A(0) = P_A(1) = 1/2 \Longrightarrow H(A) = 1 \tag{7.3}$$

The joint probability can be computed by Bayes theorem

$$P_{AB}(a, b) = P_A(a)P_{BA}(b|a) \tag{7.4}$$

One finds

$$P_{AB}(a, a) = \frac{p}{2}, \quad P_{AB}(a, \neg a) = \frac{q}{2}, \quad a \in 0, 1 \tag{7.5}$$

and $\neg$ is not a.

P_B is the marginal of P_{AB}

$$P_B(0) = P_B(1) = \frac{p+q}{2} = \frac{1}{2} \Longrightarrow H(B) = 1 \tag{7.6}$$

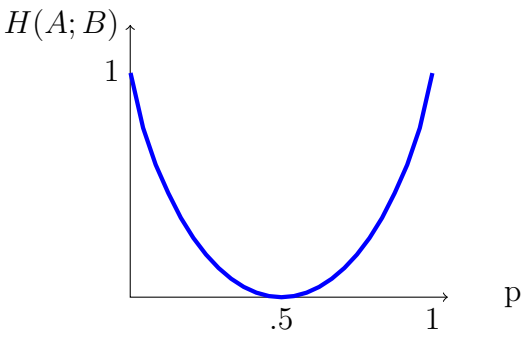
which is also clear by symmetry. The joint Shannon entropy is

$$\begin{aligned} H(A, B) &= -p \log \frac{p}{2} - q \log \frac{q}{2} \\ &= H_2(p) + 1 \end{aligned}$$

The mutual information is

$$H(A; B) = 2 - H(A, B) = 1 - H_2(p) \leq 1 \tag{7.7}$$

The mutual information says that Alice and Bob need to exchange $1/H(A; B) \geq 1$ physical bits in order to exchange one logical bit in this particular scheme. However, it does not give a protocol to do that.



The following example gives an example of a protocol:

Example 7.2 (Encoding and decodind)**.** *To fight the noise Alice encodes the logical bit in 2 bits.*

$$P_A(\mathbf{0}) = P_A(\mathbf{1}) = 1/2, \quad \mathbf{a} = aa, \quad H(A) = 1 \tag{7.8}$$

The channel acts as in the previous example

$$P_{AB}(bb'|aa) = p^2\delta_{ab}\delta_{ab'} + pq\left(\delta_{ab}\delta_{a\neq b'} + \delta_{a\neq b}\delta_{ab'}\right) + q^2\delta_{a\neq b}\delta_{a\neq b'} \tag{7.9}$$

From this you can compute P_{AB}

$$P_{AB}(aa, aa) = \frac{p^2}{2}, \;\; P_{AB}(aa, a\neg a) = P_{AB}(aa, \neg aa) = \frac{pq}{2}, \;\; P_{AB}(aa, \neg a\neg a) = \frac{q^2}{2}$$

and the marginal

$$P_B(aa) = \frac{p^2 + q^2}{2}, \quad P_B(a\neg a) = pq$$

Bob decodes the message by keeping only the identical pairs and junking the mistakes. This means he decodes the message not in bits but in trits $(0, 1, \mathbf{J})$ *The conditional probabilities are*

$$P_{AB}(\mathbf{a}|\mathbf{a}) = p^2, \quad P_{AB}(\mathbf{a}|\neg\mathbf{a}) = q^2, \quad P_{AB}(\mathbf{J}|\mathbf{a}) = pq, \quad \mathbf{a} \in \mathbf{0}, \mathbf{1}$$

From this one computes the joint probabilities

$$P_{AB}(\mathbf{a}, \mathbf{a}) = \frac{p^2}{2} \quad P_{AB}(\mathbf{a}, \mathbf{J}) = pq, \quad P_{AB}(\mathbf{a}, \neg\mathbf{a}) = \frac{q^2}{2}$$

and the marginals

$$P_B(\mathbf{0}) = P_B(\mathbf{1}) = \frac{p^2 + q^2}{2}, \quad P_B(\mathbf{J}) = 2pq$$

Now we turn to Shannon entropies. Since Bob can have 3 outcomes and jointly there are 6 outcomes

$$0 \le H(B) \le \log 3, \quad 0 \le H(A,B) \le \log 6 \tag{7.10}$$

For the case at hand

$$H(B) = (p-q)^2 - (p^2+q^2)\log(p^2+q^2) - 2pq \log pq$$

and

$$H(A,B) = (p^2+q^2) - p^2 \log p^2 - q^2 \log q^2 - 2pq \log pq$$

and mutual information

$$H(A;B) = 1 - 2pq - p^2 \log\left(1+(q/p)^2\right) - q^2 \log\left(1+(p/q)^2\right) \le 1$$

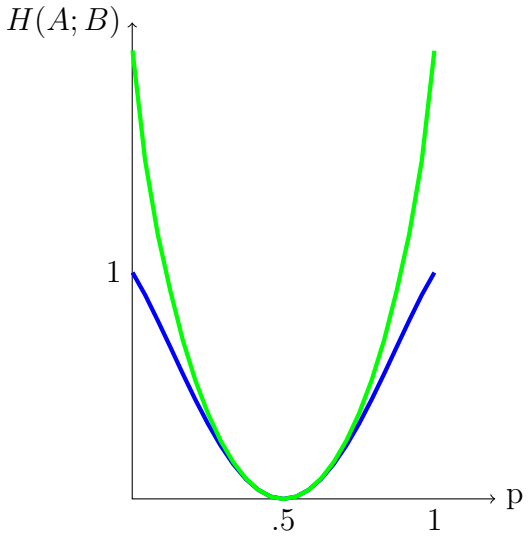


Figure 2: The blue line shows the mutual information in a pair of bits when errors are discarded. The green line is the mutual information in the previous example per two uses of the channel. It shows that the decoding and encoding we have chosen are close to optimal near $p = 1/2$ but are far from optimal near $p = 1$. For example, it may be better to encode and decode in triplets $\mathbf{a} \mapsto aaa$ with majority votes when an error occurs.

8 Shannon coding theorem

The success of communication theory is due to the fact that one can correct for errors of arbitrary long message. Even with $N \to \infty$, errors can be corrected (with high probability and provided the probability for error is not too large). Shannon coding theorem says that:

You can transmit information through a noisy channel without errors provided the rate [2] r is smaller than the channel capacity.

You might think a-priori that this must be impossible. If the message has length N and there is probability of error p for each letter, then the probability that the message is error free is exponentially small

$$p^N \to 0 \tag{8.1}$$

Indeed, you expect to have

$$pN \tag{8.2}$$

errors in the message. In fact, if the message is long enough all possible types of errors will appear. For example if you try to encode the logical bits by a triplet

$$\mathbf{0} \mapsto 000, \quad \mathbf{1} \mapsto 111 \tag{8.3}$$

and then try to fix mistakes by majority vote

$$001 \mapsto 000, \quad 010 \mapsto 000, \quad etc. \tag{8.4}$$

The method will fail when N is long enough. Indeed, the probability for two and therr consecutive bits to err in a block of three bits is

$$P(2 \text{ errors}) = 3pq^2, \quad P(\text{three errors}) = q^3 \quad q = 1 - p \tag{8.5}$$

Hence, if you send N blocks of 3 you'd expect to find

$$N(3p + q)q^2 = N(1 + 2p)q^2 \tag{8.6}$$

errors with majority voting.

How can you ever expect to be able to protect against all errors? That arbitrarily long messages can be corrected for errors if the mutual information is non-zero, was a crowning achievement of Shannon.

[2] See Rq. 10.3

9 The geometric of error correction

We can identify a binary message of length N with a vertex of the unit cube in $\mathbb{R}^N$. Here are two simple facts about the unit cube

- The total number of vertices is 2^N
- Each vertex of the has
$$\binom{N}{R} \tag{9.1}$$
neighbours at (Hamming) distance R

Hamming distance is the number of different bits in two strings of equal length.

10 Code-words

To protect against errors we pick a subset of strings as legitimate code-words. We want the Hamming distance between any two code-words is large so errors will not confuse us between codewords. An message with an error will still point to a single code-word, the one closest to it.

We pick 2^R vertices with $R < N$, chosen as far form each other as possible. These are our codewords.

Let us now compute the Hamming distance between codewords.

Each codeword is surrounded by a ball of radius P of junk vertices. The number of junk vertices per code-word is

$$2^{N-R} \tag{10.1}$$

Since in high dimensions most points of the ball lie on its surface we get that

$$N - R \approx \log \binom{N}{P} \approx N H_2(x), \quad p = \frac{P}{N} \tag{10.2}$$

This fixes the relation between R and P

$$r = 1 - H_2(p), \quad p = \frac{P}{N} \quad r = \frac{R}{N} \tag{10.3}$$

The wonderful thing about this relation is that we get an equation for the "intensive" quantities r and p, independent of N. r is a decreasing function of p (for $p < 1/2$) as one expects.

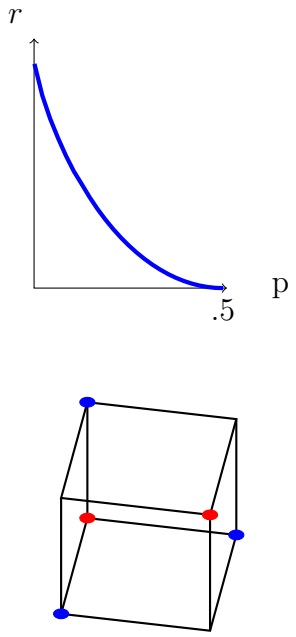


Figure 3: The vertices of the 3D cube represents the eight words of 3 bits. The two red dots $(0, 0, 0)$ and $(1, 1, 1)$ are the two code words. They are separated by Hamming distance 3. The blue dots represent errors at Hamming distance 1 from one of the code-words and can be corrected and Hamming distance 2 fro the other. These errors can be fixed.

11 Recovery from errors

If p is the probability of an error in each bit then there will be about pN errors corresponding to each code-word. Since an error would increase the Hamming distance by (at most) 1, errors in a code-word will lie in a ball of radius P

$$P = pN \tag{11.1}$$

around the code-word. It follows that for arbitrary long messages, one can still recover from errors, provided we choose the code-words sparsely enough

$$R = rN \tag{11.2}$$

with p and r as in Eq. 10.3. The larger the error rate p the fewer code-words we pick. The message will get through error free no matter how large N is.