

A note on Shannon information

<http://marekrychlik.com/node/44>

The Shannon channel coding setup

Let A be an alphabet of n arbitrary symbols s_1, s_2, \dots, s_n . We create random messages by drawing symbols from A with probabilities $P = (p_1, p_2, \dots, p_n)$. Thus, a message M of length L is a sequence of L symbols of the alphabet A , or:

$$\{ M = s_1 s_2 \dots s_L. \}$$

We ask in how many bits we can transmit this message over a noiseless channel, i.e. a channel without random errors. More precisely, we are allowed to perform any transformation on the message M resulting in a message M' which uses the alphabet $0, 1$. The only requirement is that we maintain the ability to decode an arbitrary message. Thus, the transformation $M \rightarrow M'$ must be an invertible map.

We will write $|M|$ for the length of the message. Let $H(P)$ denote the Shannon entropy of the distribution P , i.e. the number defined as follows:

$$\{ H(P) = \sum_{j=1}^n p_j \cdot (-\log_2 p_j). \}$$

One calls the quantity $I(s_j) = -\log_2 p_j$ the information contained in the event s_j . In general, if A is any event (in the sense of probability theory) then $I(A) = -\log_2 \mathbb{P}(A)$ is the amount of information contained in the fact that A occurred. We note that Shannon entropy is the mean value of the informations of the individual symbols.

Shannon's theorem gives a lower bound:

$$\{ |M'| \geq |M| \cdot H(P). \}$$

The use of the logarithm with base 2 in the above calculations is intentional if informations is to be measured in bits, and bits are used to transmit or store information. The coding algorithm

There is an algorithm to code messages arbitrarily close to the Shannon entropy bound. The method is called the Shannon-Fano method. If the probabilities are all negative powers of 2, the Huffman algorithm yields the coding algorithm which is in perfect agreement with entropy.

Let us illustrate the Huffman code for $P = (1/2, 1/4, 1/4)$ and $A = \{1, 2, 3\}$. We note that any efficient code must use short representations of frequently used symbols and long representations for rare symbols. The Huffman code is a code with this property. For our example:

- Symbol 1 is translated to 1
- Symbol 2 is translated to the sequence 00
- Symbol 3 is translated to the sequence 01

For example:

$$\{ 123321 \rightarrow 1000101001 \}$$

{We note that Huffman code has the *prefix property*, which allows us to decode Huffman encoded messages. This property says that if no symbol code is the prefix of the code for another symbol. The prefix property is not necessary to be able to invert a substitution code. However, we can see that the lack of this property makes it necessary to look ahead before decoding a symbol.

{We note that the information of each symbol coincides with the number of bits in the symbol's code. This observation automatically proves that the Huffman code realizes the bound in the Shannon theorem. *What are the savings (for the example above)?*

{The *compression ratio* is the ratio $|M|/|M'|$ and reflects the savings in space or transmission time resulting from coding of the message. In our previous example, the entropy is:

$$\{ H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right) = \frac{1}{2} \left(-\log_2 \frac{1}{2}\right) + \frac{1}{4} \left(-\log_2 \frac{1}{4}\right) + \frac{1}{4} \left(-\log_2 \frac{1}{4}\right) = \frac{1}{2} \cdot 1 + 2 \cdot \frac{1}{4} \cdot 2 = 1.5$$

{Thus, the optimal coding method should issue 1.5 bits per symbol of the message. We claim that the Huffman code described above is optimal. Indeed, a message of length L conforming to the distribution $P = (1/2, 1/4, 1/4)$ should have $L/2$ 1's, $L/4$ 2's and $L/4$ 3's. Thus, the length of the code M' can be computed directly by counting the bits:

$$\{ \frac{L}{2} \cdot 1 + \frac{L}{4} \cdot 2 + \frac{L}{4} \cdot 2 = LH(P).$$

{We note that $H(P) = 1.5$. Thus the gain of using the bias of the distribution $(1/2, 1/4, 1/4)$ towards 1 vs. the unbiased distribution $(1/3, 1/3, 1/3)$ is:

$$\{ \frac{\log_2 3}{1.5} \approx 1.05664$$

{This amounts to about 5 percent, but it proves the principle.

{Indeed, much more significant savings are possible, and they are fundamentally important to such technological achievements as digital television. *We note that the bit rate of $\log_2 3$ for the distribution $(1/3, 1/3, 1/3)$ can be realized by a technique called *arithmetic coding*.*