Tradeoffs Choosing Codeword Lengths

The Kraft-McMillan inequalities imply that to make some codewords shorter, we will have to make others longer.

Example: The obvious binary encoding for eight symbols uses codewords that are all three bits long. This code is instantaneous, and satisfies the Kraft inequality, since:

$$\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3} \;=\; 1$$

Suppose we want to encode the first symbol using only two bits. We'll have to make some other codewords longer – eg, we can encode two of the other symbols in four bits, and the remaining five symbols in three bits, since

$$\frac{1}{2^2}+\frac{1}{2^4}+\frac{1}{2^4}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3}+\frac{1}{2^3} \;=\; 1$$

How should we choose among the possible codes?

Formalizing Which Codes are the Best: Probabilities for Source Symbols

We'd like to choose a code that uses short codewords for common symbols and long ones for rare symbols.

To formalize this, we need to assign each symbol in the source alphabet a probability. Symbols $a_1, \ldots, a_I$ will have probabilities written as $p_1, \ldots, p_I$. We assume that these probabilities don't change with time.

We also assume that symbols in the source sequence, X_1, X_2, ..., X_N, are *independent*:

$$\begin{aligned} P(X_1 = a_{i_1},\ X_2 = a_{i_2},\ \ldots,\ X_n = a_{i_N}) \\ = \; p_{i_1} p_{i_2} \cdots p_{i_N} \end{aligned}$$

These assumptions are really too restrictive in practice, but we'll ignore that for now.

Expected Codeword Length

Consider a code whose codewords for symbols $a_1, \ldots, a_I$ have lengths $l_1, \ldots, l_I$. Let the probabilities of these symbols be $p_1, \ldots, p_I$. We define the *expected codeword length* for this code to be

$$L \;=\; L(C, X) \;=\; \sum_{i=1}^{I} p_i l_i$$

This is the average length of the codeword encoding a single source symbol. But since averaging is a linear operation, the average length of a coded message with N source symbols is just NL. For example, when $N=3$:

$$\begin{aligned} &\sum_{i_1=1}^{I} \sum_{i_2=1}^{I} \sum_{i_3=1}^{I} p_{i_1} p_{i_2} p_{i_3} (l_{i_1} + l_{i_2} + l_{i_3}) \\ &= \sum_{i_1=1}^{I} p_{i_1} l_{i_1} + \sum_{i_2=1}^{I} p_{i_2} l_{i_2} + \sum_{i_3=1}^{I} p_{i_3} l_{i_3} \;=\; 3L \end{aligned}$$

We aim to choose a code for which L is small.

Optimal Codes

We say a code is *optimal* for a given source (with given symbol probabilities) if its average length is at least as small as that of any other code.

There can be many optimal codes for the same source, all with the same average length.

The Kraft-McMillan inequalities imply that if there is an optimal code, there is also an optimal *instantaneous* code. More generally, for any uniquely decodable code with average length L, there is an instantaneous code with the same average length.

Questions: Can we figure out the length of an optimal code from the symbol probabilities? Can we find such an optimal code, and use it in practice?

Shannon Information Content

A plausible proposal:

The *amount of information* obtained when we learn that $X = a_i$ is $\log_2(1/p_i)$ bits, where $p_i = P(X = a_i)$.

Example:

We learn which of 64 equally-likely possibilities has occurred. The Shannon information content is $\log_2(64) = 6$ bits. This makes sense, since we could encode the result using codewords that are all 6 bits long, and we have no reason to favour one symbol over another by using a code of varying length.

For further intuitions about why this is a plausible measure of information, see Section 4.1 of MacKay's book.

The Entropy of a Source

The Shannon information content pertains to a single value of the random variable X. To find out how much information learning the value of X conveys *on average*, we find the expected value of the Shannon information content.

This is called the *entropy* of the random variable (or source), and is symbolized by H:

$$H(X) \;=\; \sum_{i=1}^{I} p_i \log_2(1/p_i)$$

where $p_i = P(X = a_i)$.

When the logarithm is to base 2, as above, the entropy has units of bits. (We could use some other base; when base e is used, the units are called "nats".)

Information, Entropy, and Codes

How does this relate to data compression?

A vague idea: Since receipt of symbol a_i conveys $\log_2(1/p_i)$ bits of "information", this symbol "ought" to be encoded using a codeword with that many bits. Problem: $\log_2(1/p_i)$ isn't always an integer.

A consequence: If this is done, then the expected codeword length will be equal to the entropy: $\sum_{i=1}^{I} p_i \log_2(1/p_i)$.

A vague conjecture: The expected codeword length for an optimal code ought to be equal to the entropy.

But it's easy to see that this can't quite be right as stated. Consider $p_0 = 0.1$, $p_1 = 0.9$, so $H = 0.469$. But the optimal code for a symbol with only two values obviously uses codewords 0 and 1, with expected length of 1.

A Property of the Entropy

For any two probability distributions, $p_1, \ldots, p_I$ and $q_1, \ldots, q_I$:

$$\sum_{i=1}^{I} p_i \log_2\left(\frac{1}{p_i}\right) \;\le\; \sum_{i=1}^{I} p_i \log_2\left(\frac{1}{q_i}\right)$$

Proof:

First, note that for all $x > 0$, $\ln x \le x-1$. So $\log_2 x \le (x-1)/\ln 2$.

We can now show that the LHS-RHS above is:

$$\sum_{i=1}^{I} p_i \left[\log_2\left(\frac{1}{p_i}\right) - \log_2\left(\frac{1}{q_i}\right)\right] \;=\; \sum_{i=1}^{I} p_i \log_2\left(\frac{q_i}{p_i}\right)$$

$$\le\; \frac{1}{\ln 2} \sum_{i=1}^{I} p_i \left(\frac{q_i}{p_i} - 1\right) \;=\; \frac{1}{\ln 2}\left(\sum_{i=1}^{I} q_i - \sum_{i=1}^{I} p_i\right) \;=\; 0$$

Proving We Can't Compress to Less Than the Entropy

We can use this result to prove that any uniquely decodable binary code for X must have expected length of at least $H(X)$:

Proof:

Let the codeword lengths be $l_1, \ldots, l_I$, and define $K = \sum_{i=1}^{I} 2^{-l_i}$ and $q_i = 2^{-l_i}/K$.

The q_i can be seen as probabilities, so

$$H(X) \;=\; \sum_{i=1}^{I} p_i \log_2\left(\frac{1}{p_i}\right) \;\le\; \sum_{i=1}^{I} p_i \log_2\left(\frac{1}{q_i}\right)$$

$$=\; \sum_{i=1}^{I} p_i \log_2(2^{l_i}K) \;=\; \sum_{i=1}^{I} p_i(l_i + \log_2 K)$$

Since the code is uniquely decodable, $K \le 1$ and hence $\log_2 K \le 0$. From this, we can conclude that $\sum p_i l_i \ge H(X)$.

Shannon-Fano Codes

If we can't choose codewords with the "right" lengths, $\log_2(1/p_i)$, we can try to get close.

Shannon-Fano codes are constructed so that the codewords for the symbols, with probabilities $p_1, \ldots, p_I$, have lengths

$$l_i \;=\; \lceil \log_2(1/p_i) \rceil$$

Here, $\lceil x \rceil$ is the smallest integer greater than or equal to x.

The Kraft inequality says such a code exists, since

$$\sum_{i=1}^{I} \frac{1}{2^{l_i}} \;\le\; \sum_{i=1}^{I} \frac{1}{2^{\log_2(1/p_i)}} \;=\; \sum_{i=1}^{I} p_i \;=\; 1$$

Example:

p_i:	0.4	0.3	0.2	0.1
$\log_2(1/p_i)$:	1.32	1.74	2.32	3.32
$l_i = \lceil \log_2(1/p_i) \rceil$:	2	2	3	4
Codeword:	00	01	100	1100

Expected Lengths of Shannon-Fano Codes

The expected length of a Shannon-Fano code for X, if symbols have probabilities $p_1, \ldots, p_I$, is

$$\begin{aligned}
\sum_{i=1}^{I} p_i l_i \;&=\; \sum_{i=1}^{I} p_i \lceil \log_2(1/p_i) \rceil \\
&<\; \sum_{i=1}^{I} p_i\,(1 + \log_2(1/p_i)) \\
&=\; \sum_{i=1}^{I} p_i \;+\; \sum_{i=1}^{I} p_i \log_2(1/p_i)) \\
&=\; 1 + H(X)
\end{aligned}$$

This gives an *upper bound* on the expected length of an optimal code for X. However, the Shannon-Fano code itself may not be optimal (though it sometimes is).

What Have We Shown?

We've now proved the following (Theorem 5.1 in MacKay's book):

> A source X can be encoded using an instantaneous code, C, with expected length, $L(C,X)$, satisfying
>
> $$H(X) \;\le\; L(C,X) \;<\; H(X) + 1$$

Two main theoretical problems remain:

- Can we find *optimal* codes, which actually minimize L?
- Can we somehow close the gap between $H(X)$ and $H(X)+1$ above, to show that the entropy is the exactly correct way of measuring the average information content of a source?