Shannon Uncertainty and Information

Alan R. Rogers

April 8, 2019

William Dembski [1] defines a measure of information that is similar to the classical measure of Claude Shannon [2]. To understand Dembski, you need a little background on Shannon. This note provides that background.

Suppose that we are transmitting messages, each of which consists of a single letter: either a, b, or c. We wish to transmit such messages over a communications channel that uses only 0s and 1s. Each 0 and 1 is called a "binary digit" or "bit." To reduce the load on our communications channel, our code should use (on average) the minimum possible number of bits. Suppose that the three messages, a, b, and c, occur with probabilities 1/2, 1/4, and 1/4. Here is one possible code:

Message	Probability	Code
a	1/2	0
b	1/4	10
c	1/4	11

We use 1 bit for the most common message and 2 each for the less-common ones. Half the messages use 1 bit and half use 2, so the average message uses 1.5. This can be thought of a measure of "uncertainty."

Suppose now that we have received a message, but it is slightly garbled. It wasn't "a," but was it "b" or "c?" We're completely undecided. We can use a simpler code to represent the remaining uncertainty:

Message	Probability	Code
b	1/2	0
c	1/2	1

After receiving the slightly-garbled message, only a single bit is needed to represent the remaining uncertainty. The reduction in uncertainty—0.5 bits—is the *Shannon information* provided by the message.

Shannon [2] developed a theory of uncertainty and information. He showed that the uncertainty cannot be smaller than

$$H = -\sum p_i \log_2 p_i,$$

where p_i the the probability of the ith message, and $\log_2 p_i$ is the logarithm of p_i to base 2. (See the appendix for a refresher on logarithms.) This is the "Shannon uncertainty." Before receiving the garbled message in the example above, the Shannon uncertainty was

$$\begin{aligned} H_0 &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} \\ &= \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 1.5, \end{aligned}$$

After receiving the garbled message, the Shannon uncertainty is

$$H_1 = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

In both cases, Shannon's formula gives the answer that we got by counting bits. This is not typical. As we shall see, most codes do not quite achieve Shannon's theoretical minimum.

Finally, suppose that we received an ungarbled message. After receiving it, we know with certainty that the letter is "b." There is no further uncertainty, so $H_1 = 0$. In terms of Shannon's formula,

$$H_1 = -1 \times \log_2 1 = 0$$

because the logarithm of 1 is 0. The information provided by the ungarbled message, $H_0 - H_1 = 1.5 - 0 = 1.5$ bits, is equal to the uncertainty of the system before we received the message.

In the examples above, we get the same answer two different ways—from Shannon's formula and by averaging the number of bits per message. This is not typical. Shannon's formula usually gives a slightly smaller answer than the best code you can cook up. For example, consider:

Message	Frequency	Code
a	3/5	0
b	1/5	10
c	1/5	11

The average number of bits per message is

$$\frac{3}{5} \times 1 + \frac{2}{5} \times 2 = 1.4$$

Shannon's formula gives a slightly smaller number,

$$\begin{aligned} H &= -\frac{3}{5}\log_2\frac{3}{5} - \frac{1}{5}\log_2\frac{1}{5} - \frac{1}{5}\log_2\frac{1}{5} \\ &= 0.442 + 0.464 + 0.464 \\ &= 1.37 \end{aligned}$$

The discrepancy arises because in Shannon's formula, the number of bits allotted to a message is $-\log_2 p$ bits, which may be a fraction. In the example just above, message "a" had probability 3/5, so Shannon's formula allots only a fraction, $-\log_2(3/5) = 0.74$, of a bit. Our code, however, used an entire bit. Shannon's formula is a lower bound on the number of bits.

Appendix: logarithms

We all know that

$$2^3 = 2 \times 2 \times 2 = 8$$

This same fact can also be expressed by writing

$$\log_2 8 = 3$$

In words, this reads "the logarithm to base 2 of 8 equals 3." Both equations mean the same thing: the 3rd power of 2 is 8. Similarly, $2^{10} = 1024$, so $\log_2 1024 = 10$.

For negative exponents, the situation is similar:

$$2^{-3} = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

We write the same fact in logarithms as

$$\log_2 \frac{1}{8} = -3$$

Here are some other numbers and their base-2 logs:

x	$\log_2 x$
1/4	–2
1/2	–1
1	0
2	1
4	2
8	3
16	4
1,099,511,627,776	40

As x increases, $\log_2 x$ increases too, but much more slowly.

References

[1] William A. Dembski. *The Design Inference: Eliminating Chance through Small Probabilities*. Cambridge University Press, 1998.

[2] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.