

# SOURCE CODING THEOREM\*

Don Johnson

This work is produced by OpenStax-CNX and licensed under the  
Creative Commons Attribution License 1.0<sup>†</sup>

## Abstract

The Source Coding Theorem states that the entropy of an alphabet of symbols specifies to within one bit how many bits on the average need to be used to send the alphabet.

The significance of an alphabet's entropy rests in how we can represent it with a sequence of **bits**. Bit sequences form the "coin of the realm" in digital communications: they are the universal way of representing symbolic-valued signals. We convert back and forth between symbols to bit-sequences with what is known as a **codebook**: a table that associates symbols to bit sequences. In creating this table, we must be able to assign a **unique** bit sequence to each symbol so that we can go between symbol and bit sequences without error.

**POINT OF INTEREST:** You may be conjuring the notion of hiding information from others when we use the name codebook for the symbol-to-bit-sequence table. There is no relation to cryptology, which comprises mathematically provable methods of securing information. The codebook terminology was developed during the beginnings of information theory just after World War II.

As we shall explore in some detail elsewhere, digital communication is the transmission of symbolic-valued signals from one place to another. When faced with the problem, for example, of sending a file across the Internet, we must first represent each character by a bit sequence. Because we want to send the file quickly, we want to use as few bits as possible. However, we don't want to use so few bits that the receiver cannot determine what each character was from the bit sequence. For example, we could use one bit for every character: File transmission would be fast but useless because the codebook creates errors. Shannon<sup>1</sup> proved in his monumental work what we call today the **Source Coding Theorem**. Let  $B(a_k)$  denote the number of bits used to represent the symbol  $a_k$ . The average number of bits  $\bar{B}(A)$  required to represent the entire alphabet equals  $\sum_{k=1}^K B(a_k) Pr[a_k]$ . **The Source Coding Theorem states that the average number of bits needed to accurately represent the alphabet need only to satisfy**

$$H(A) \leq \bar{B}(A) < H(A) + 1 \quad (1)$$

Thus, the alphabet's entropy specifies to within one bit how many bits on the average need to be used to send the alphabet. The smaller an alphabet's entropy, the fewer bits required for digital transmission of files expressed in that alphabet.

---

\*Version 2.14: Jun 27, 2010 5:45 pm -0500

<sup>†</sup> <http://creativecommons.org/licenses/by/1.0>

<sup>1</sup> <http://www.lucent.com/minds/inftheory/>

**Example 1**

A four-symbol alphabet has the following probabilities.

$$Pr[a_0] = \frac{1}{2}$$

$$Pr[a_1] = \frac{1}{4}$$

$$Pr[a_2] = \frac{1}{8}$$

$$Pr[a_3] = \frac{1}{8}$$

and an entropy of 1.75 bits. Let's see if we can find a codebook for this four-letter alphabet that satisfies the Source Coding Theorem. The simplest code to try is known as the **simple binary code**: convert the symbol's index into a binary number and use the same number of bits for each symbol by including leading zeros where necessary.

$$a_0 \leftrightarrow 00a_1 \leftrightarrow 01a_2 \leftrightarrow 10a_3 \leftrightarrow 11 \tag{2}$$

Whenever the number of symbols in the alphabet is a power of two (as in this case), the average number of bits  $\bar{B}(A)$  equals  $\log_2 K$ , which equals 2 in this case. Because the entropy equals 1.75 bits, the simple binary code indeed satisfies the Source Coding Theorem—we are within one bit of the entropy limit—but you might wonder if you can do better. If we chose a codebook with differing number of bits for the symbols, a smaller average number of bits can indeed be obtained. The idea is to use shorter bit sequences for the symbols that occur more often. One codebook like this is

$$a_0 \leftrightarrow 0a_1 \leftrightarrow 10a_2 \leftrightarrow 110a_3 \leftrightarrow 111 \tag{3}$$

Now  $\bar{B}(A) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = 1.75$ . We can reach the entropy limit! The simple binary code is, in this case, less efficient than the unequal-length code. Using the efficient code, we can transmit the symbolic-valued signal having this alphabet 12.5% faster. Furthermore, we know that no more efficient codebook can be found because of Shannon's Theorem.