Source Coding*

Behnaam Aazhang

Abstract

An introduction to the concept of typical sequences, which lie at the heart of source coding. The idea of typical sequences leads to Shannon's source-coding Theorem.

As mentioned earlier, how much a source can be compressed should be related to its entropy. In 1948, Claude E. Shannon introduced three theorems and developed very rigorous mathematics for digital communications. In one of the three theorems, Shannon relates entropy to the minimum number of bits per second required to represent a source without much loss (or distortion).

Consider a source that is modeled by a discrete-time and discrete-valued random process X_1, X_2, ..., X_n, ... where $x_i \in \{a_1, a_2, \ldots, a_N\}$ and define $p_{X_i}(x_i = a_j) = p_j$ for $j = 1, 2, \ldots, N$, where it is assumed that X_1, X_2,... X_n are mutually independent and identically distributed.

Consider a sequence of length n

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \tag{1}$$

The symbol a_1 can occur with probability p_1. Therefore, in a sequence of length n, on the average, a_1 will appear np_1 times with high probabilities if n is very large.

Therefore,

$$P(X = x) = p_{X_1}(x_1)\, p_{X_2}(x_2) \ldots p_{X_n}(x_n) \tag{2}$$

$$P(X = x) \simeq {p_1}^{np_1} {p_2}^{np_2} \ldots {p_N}^{np_N} = \prod_{i=1}^{N} {p_i}^{np_i} \tag{3}$$

where $p_i = P(X_j = a_i)$ for all j and for all i.

*Version 2.10: Oct 3, 2005 1:45 pm -0500

†http://creativecommons.org/licenses/by/1.0

A typical sequence X may look like

$$X = \begin{pmatrix} a_2 \\ \vdots \\ a_1 \\ a_N \\ a_2 \\ a_5 \\ \vdots \\ a_1 \\ \vdots \\ a_N \\ a_6 \end{pmatrix} \tag{4}$$

where a_i appears np_i times with large probability. This is referred to as a **typical sequence**. The probability of X being a typical sequence is

$$\begin{aligned} P(X = x) \simeq \prod_{i=1}^{N} {p_i}^{np_i} &= \prod_{i=1}^{N} \left(2^{\log_2 p_i}\right)^{np_i} \\ &= \prod_{i=1}^{N} 2^{np_i \log_2 p_i} \\ &= 2^{n \sum_{i=1}^{N} p_i \log_2 p_i} \\ &= 2^{-(nH(X))} \end{aligned} \tag{5}$$

where $H(X)$ is the entropy of the random variables X_1, X_2,..., X_n.

For large n, almost all the output sequences of length n of the source are equally probably with probability $\simeq 2^{-(nH(X))}$. These are typical sequences. The probability of nontypical sequences are negligible. There are N^n different sequences of length n with alphabet of size N. The probability of typical sequences is almost 1.

$$\sum_{k=1}^{\text{\# of typical seq.}} 2^{-(nH(X))} = 1 \tag{6}$$

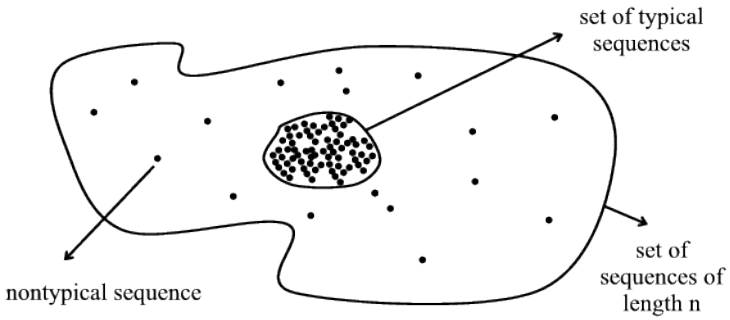

Figure 1

Example 1

Consider a source with alphabet {A,B,C,D} with probabilities { $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$}. Assume X_1, X_2,..., X_8 is an independent and identically distributed sequence with $X_i \in \{A, B, C, D\}$ with the above probabilities.

$$\begin{aligned} H(X) &= \left(-\left(\tfrac{1}{2}\log_2\tfrac{1}{2}\right)\right) - \tfrac{1}{4}\log_2\tfrac{1}{4} - \tfrac{1}{8}\log_2\tfrac{1}{8} - \tfrac{1}{8}\log_2\tfrac{1}{8} \\ &= \tfrac{1}{2} + \tfrac{2}{4} + \tfrac{3}{8} + \tfrac{3}{8} \\ &= \tfrac{4+4+6}{8} \\ &= \tfrac{14}{8} \end{aligned} \tag{7}$$

The number of typical sequences of length 8

$$2^{8\times\frac{14}{8}} = 2^{14} \tag{8}$$

The number of nontypical sequences $4^8 - 2^{14} = 2^{16} - 2^{14} = 2^{14}(4-1) = 3\times 2^{14}$

Examples of typical sequences include those with A appearing $8\times\frac{1}{2} = 4$ times, B appearing $8\times\frac{1}{4} = 2$ times, *etc.* {A,D,B,B,A,A,C,A}, {A,A,A,A,C,D,B,B} and much more.

Examples of nontypical sequences of length 8: {D,D,B,C,C,A,B,D}, {C,C,C,C,C,B,C,C} and much more. Indeed, these definitions and arguments are valid when n is very large. The probability of a source output to be in the set of typical sequences is 1 when $n \to \infty$. The probability of a source output to be in the set of nontypical sequences approaches 0 as $n \to \infty$.

The essence of source coding or data compression is that as $n \to \infty$, nontypical sequences never appear as the output of the source. Therefore, one only needs to be able to represent typical sequences as binary codes and ignore nontypical sequences. Since there are only $2^{nH(X)}$ typical sequences of length n, it takes $nH(X)$ bits to represent them on the average. On the average it takes $H(X)$ bits per source output to represent a simple source that produces independent and identically distributed outputs.

Theorem 1: Shannon's Source-Coding

A source that produced independent and identically distributed random variables with entropy H can be encoded with arbitrarily small error probability at any rate R in bits per source output if $R \geq H$. Conversely, if $R < H$, the error probability will be bounded away from zero, independent of the complexity of coder and decoder.

The source coding theorem proves existence of source coding techniques that achieve rates close to the entropy but does not provide any algorithms or ways to construct such codes.

If the source is not i.i.d. (independent and identically distributed), but it is stationary with memory, then a similar theorem applies with the entropy $H(X)$ replaced with the entropy rate $H = \lim_{n\to\infty} H(X_n|X_1X_2\ldots X_{n-1})$

In the case of a source with memory, the more the source produces outputs the more one knows about the source and the more one can compress.

Example 2

The English language has 26 letters, with space it becomes an alphabet of size 27. If modeled as a memoryless source (no dependency between letters in a word) then the entropy is $H(X) = 4.03$ bits/letter.

If the dependency between letters in a text is captured in a model the entropy rate can be derived to be $H = 1.3$ bits/letter. Note that a non-information theoretic representation of a text may require 5 bits/letter since 2^5 is the closest power of 2 to 27. Shannon's results indicate that there may be a compression algorithm with the rate of 1.3 bits/letter.

Although Shannon's results are not constructive, there are a number of source coding algorithms for discrete time discrete valued sources that come close to Shannon's bound. One such algorithm is the Huffman source coding algorithm. Another is the Lempel and Ziv algorithm.

Huffman codes and Lempel and Ziv apply to compression problems where the source produces discrete time and discrete valued outputs. For cases where the source is analog there are powerful compression algorithms that specify all the steps from sampling, quantizations, and binary representation. These are referred to as waveform coders. JPEG, MPEG, vocoders are a few examples for image, video, and voice, respectively.