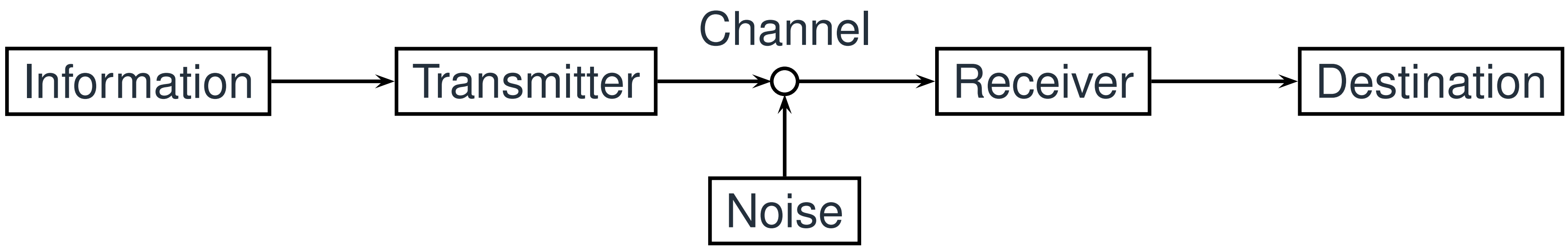
Information Theory

Information theory starts with a classic paper by Claude Shannon in 1948 entitled *A mathematical theory of communication*.



- Messages normally have *meaning* (information).
- The actual message is one selected from a set of possible messages.
- The communication system must operate for all possible messages, not just the one we send.

Information theory

Information about an event is closely related to its probability of occurrence.

- An event which is certain (100% probability) contains no information
- A high probability of occurrence contains little information
- A rare occurrence contains relatively large amounts of information

For example, on a roulette wheel a winning inside bet of a single number (probability 1/37) conveys more information than an outside bet, such as *odd numbers* (probability 18/37).

The information I_A associated with an event A occuring with probability P_A

$$I_A = \log \frac{1}{P_A} = -\log P_A$$

For a binary (digital) system we use base 2, i.e. $\log_2$, giving information in *bits*

DNA example

- DNA is a molecule which contains the genetic information for life. The information is stored as a sequence of 4 possible nucleotides (bases): Adenine (A), Cytosine (C), Thynine (T) and Guanine (G), all with equal probability $P_i = 1/4$.
- The complete genome (all the genetic DNA) for E coli (a form of bacteria) has 4×10^6 of these bases.What is the information content?

$$I_{\text{E Coli}} = 4 \times 10^6 \times \log_2 4 = 8 \times 10^6 \text{ bit}$$

- The complete genome for a human being has approximately 3.2×10^9 of these bases. What is the information content?

$$I_{\text{human}} = 3.2 \times 10^9 \times \log_2 4 = 6.4 \times 10^9 \text{ bit}$$

a single-layer DVD-ROM has a capacity of 4.7 GB=38 $\times$ 10^9 bit

Source Entropy

Suppose we have some information source which uses symbols S_i with a probability P_i from an alphabet. Each symbol is chosen independently with no memory of any previous choice. We call this a discrete memoryless source (DMS).

The weighted average information per event for N possible events is given by

$$H = \sum_{i=0}^{N-1} P_i \log_2 \frac{1}{P_i}$$

We call this *entropy* in analogy with its use in thermodynamics as a measure of randomness in a system. Note that for a complete set $\sum_{i=0}^{N-1} P_i = 1$.

Source Entropy

Example: the 4 symbols A, B, C and D form a complete set occurring with probabilities $1/2$, $1/4$, $1/8$ and $1/8$ respectively.

1. The information in the 3 symbol message $X = BDA$ (assuming that the symbols are statistically independent) is

$$I_X = \log_2 4 + \log_2 8 + \log_2 2 = 2 + 3 + 1 = 6 \text{ bit}$$

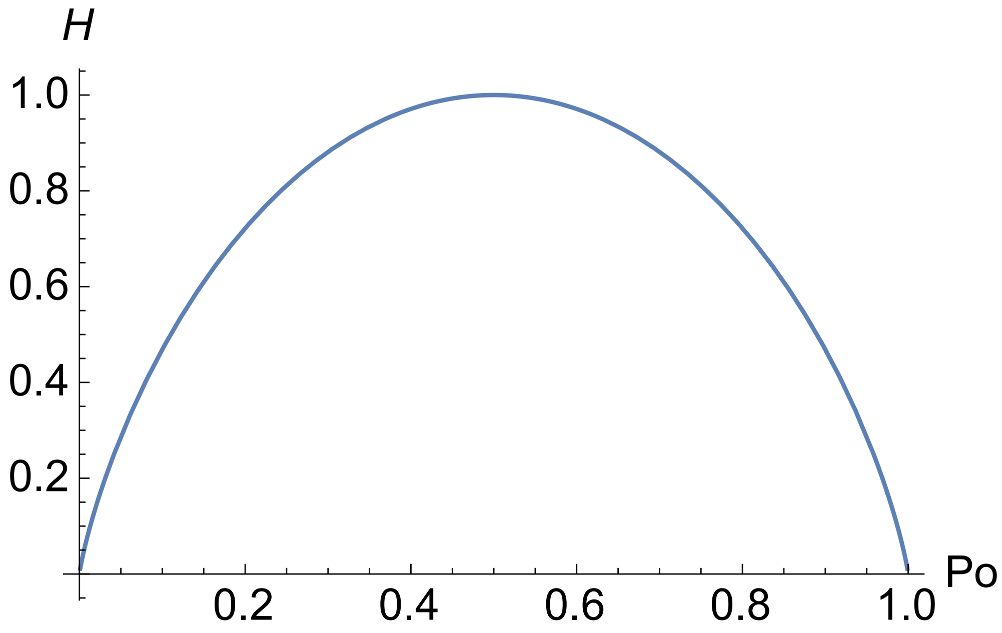
2. The source entropy for this set is

$$H = \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{8}\log_2 8 + \frac{1}{8}\log_2 8 = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{3}{8} = 1.75 \text{ bit}$$

Entropy for a Binary Source

A binary system has two symbols “0” and “1”. If the probability of a “0” symbol is P_0, then the probability of the “1” symbol $P_1 = (1 - P_0)$ and the source entropy is

$$H = -P_0 \log_2 P_0 - (1 - P_0) \log_2 (1 - P_0)$$



Source-Coding Theorem

The aim of source coding is to take the source data and make it smaller.

The source-coding theorem establishes a fundamental limit on the rate at which the output of an information source can be compressed without causing a large error probability.

- A source with entropy rate H can be encoded with arbitrarily small error probability at any rate R (bits/source output) as long as $R > H$.
- Conversely if $R < H$, the error probability will be bounded away from zero, independent of the complexity of the encoder and the decoder employed.

Huffman coding

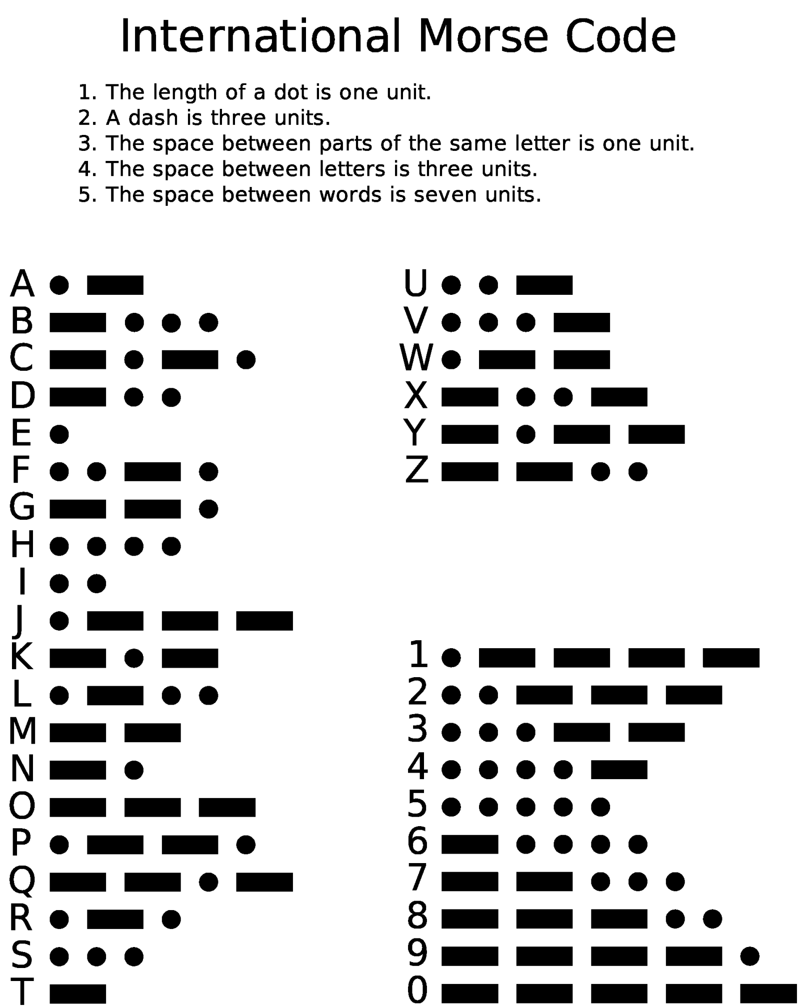
Huffman code is a particular type of optimal prefix code that is commonly used for lossless data compression.

1. sort source outputs in decreasing order of probability
2. merge two least probable into a single output is sum of their probabilities
3. if the number of remaining outputs is > 2 then go to step 1
4. arbitrarily assign 0 and 1 as code words for 2 remaining outputs
5. if an output corresponds to a merged output, append the current code word with 0 or 1 then repeat until all original outputs are assigned a code word

Example: the 4 symbols A, B, C and D form a complete set occurring with probabilities $1/2$, $1/4$, $1/8$ and $1/8$ respectively.

We get the same result using probabilities 0.4, 0.3, 0.2, 0.1, but the efficiency increases to 1.9 bit per symbol as the probabilities are no longer the ideal 2^{-n}.

Morse Code



Most commonly used symbols in the English language use the shortest codes.

Alfred Vail estimated the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown.

Lempel-Ziv-Welch (LZW) coding

Lempel-Ziv-Welch (LZW) is a universal lossless data compression algorithm. It avoids the problem in the Huffman coding of needing to know the source probabilities in advance.

A high level view of the encoding algorithm is shown here:

1. Initialize the dictionary to contain all strings of length one.
2. Find the longest string W in the dictionary that matches the current input.
3. Emit the dictionary index for W to output and remove W from the input.
4. Add W followed by the next symbol in the input to the dictionary.
5. Go to Step 2.

Decoding is fairly straightforward as the decoder builds the same extended dictionary as the encoder.

Lempel-Ziv-Welch (LZW) coding

example: she_sells_sea_shells_on_the_sea_shore

First start with an initial dictionary where _ $\rightarrow$ 0 and a–z $\rightarrow$ 1–26 (5 bit).

char	s	h	e	_	s	e	l	l	s	_s	e	a	_s	he
code	19	8	5	0	19	5	12^*	12	19	30	5	1	30	28
extended	sh	he	e_	_s	se	el	ll	ls	s_	_se	ea	a_	_sh	hel
dictionary	27	28	29	30	31	32	33	34	35	36	37	38	39	40

char	ll	s_	o	n	_	t	he	_se	a_	sh	o	r	e
code	33	35	15	14	0	20	28	36	38	27	15	18	5
extended	lls	s_o	on	n_	_t	th	he_	_sea	a_s	sho	or	re	
dictionary	41	42	43	44	45	46	47	48	49	50	51	52	

original $37 \times 5\,\text{bit} = 185\,\text{bit}$

compressed $6 \times 5\,\text{bit} + 21 \times 6\,\text{bit} = 156\,\text{bit}$

a 16% reduction. Bigger reductions for longer strings: text files typically $\sim$50%.

Data Compression

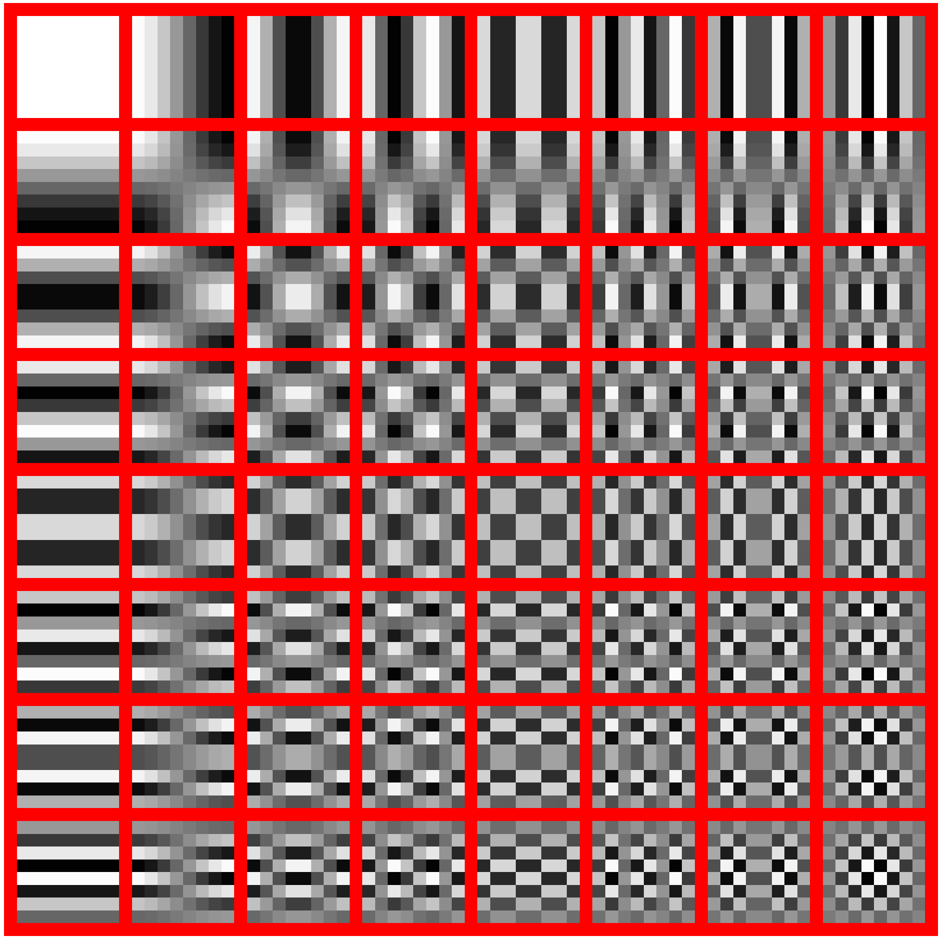
Lossless data compression algorithms usually exploit statistical redundancy to represent data without losing any information, so that the process is reversible

LZW GIF image files
DEFLATE PKZIP, Gzip, and PNG images
LZR Zip
LZX Microsoft's CAB format
FLAC audio coding format for lossless compression of digital audio

Lossy compression reduces bits by removing unnecessary or less important information (irreversible).

JPEG image files
MP3 or Vorbis audio files and streams
MPEG-2 Part 2 Video, e.g. DVD Video, Blu-ray, Digital Video Broadcasting
MPEG-4 AVC Video, e.g. Blu-ray, HD DVD, Digital Video Broadcasting

JPEG lossy compression



Discrete cosine transform