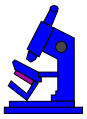



Chapter 4

Lecture 4 - Information theory

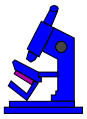

Information theory

The term ***information*** is commonly understood. Consider the following two sentences:

1. The sun will rise tomorrow.

2. The Fiji rugby team will win against the All Blacks (New Zealand rugby team) the next time they play.

Question: Which sentence contains the most information?



Information theory

✘ Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, Temperature today is OK, ...

... total information here is close to zero!

?

More information means less predictable

Less information means more predictable

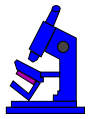
Information theory

Nyquist (1924) and Hartley (1928) laid the foundations:

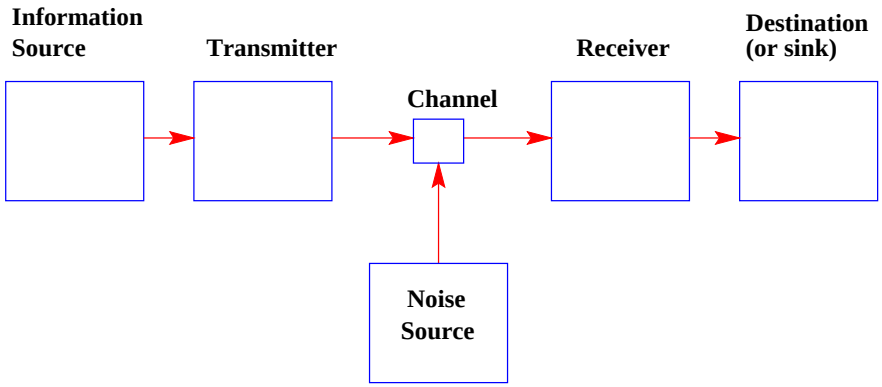
✔ Hartley showed that the information content is proportional to the *logarithm* of the number of possible messages. Integers between 1 and n need $\log_2 n$ bits.

✔ Shannon developed a mathematical treatment of communication and information in an important paper at

http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html



Information theory model



The relevance of Shannon to secrecy is in another important paper at

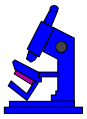
http://www.cs.ucla.edu/~jkong/research/security/shannon.html

Entropy

In our communication model, the units of transmission are called *messages*, constructed from an alphabet of (say) n symbols $x \in \{x_1, \ldots, x_n\}$ each with a probability of transmission P_x.

We associate with each symbol x a quantity H_x which is a measure of the *information* associated with that symbol.

$$H_x = P_x \log_2 \frac{1}{P_x}$$



Entropy

$$H_x = P_x \log_2 \frac{1}{P_x}$$

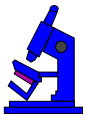
If the probability of occurence of each symbol is the same, we can derive Hartley's result, that the average amount of information transmitted in a single symbol (the source *entropy*) is

$$H(X) = \log_2 n$$

where X is a label referring to each of the source symbols $x_1, \ldots, x_n$.

Entropy units

Our units for entropy can be *bits/second* or *bits/symbol*, and we also sometimes use unit-less *relative* entropy measures (relative to the entropy of the system if all symbols were equally likely).



Entropy - same probability

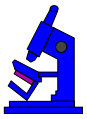
Symbols	Entropy of each symbol	Bits needed
2	$H_x = \frac{1}{2}\log_2 2 = \frac{1}{2}$	$2 * \frac{1}{2} = 1$
4	$H_x = \frac{1}{4}\log_2 4 = \frac{1}{2}$	$4 * \frac{1}{2} = 2$
8	$H_x = \frac{1}{8}\log_2 8 = \frac{3}{8}$	$8 * \frac{3}{8} = 3$
16	$H_x = \frac{1}{16}\log_2 16 = \frac{4}{16}$	$16 * \frac{4}{16} = 4$
21	$H_x = \frac{1}{21}\log_2 21 = \frac{4.39}{21}$	$21 * \frac{4.39}{21} = 4.39$

Entropy - different probability

However, if the probability of occurence of each symbol is not the same, we derive the following result, that the source *entropy* is

$$H(X) = \sum_{i=1}^{n} P_{x_i} \log_2 \frac{1}{P_{x_i}}$$

Shannon's paper shows that H determines the channel capacity required to transmit the desired information with the *most* efficient coding scheme.



Entropy - different probability

If we had a source emitting two symbols, 0 and 1, with probabilities of 1 and 0, then the entropy of the source is

$$\begin{aligned} H(X) &= \sum_{i=1}^{n} P_{x_i} \log_2 \frac{1}{P_{x_i}} \\ &= 1 * \log_2 1 + 0 * \log_2 \frac{1}{0} \\ &= 0 \text{ bits/symbol} \end{aligned}$$

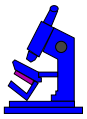
Note that

$$\lim_{y \to 0} y \log_2 \frac{1}{y} = 0$$

Entropy - different probability

If we were transmitting a sequence of letters *A,B,C,D,E* and *F* with probabilities $\frac{1}{2}$,$\frac{1}{4}$,$\frac{1}{16}$,$\frac{1}{16}$,$\frac{1}{16}$ and $\frac{1}{16}$, the entropy for the system is

$$\begin{aligned} H(X) &= \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{4}{16}\log_2 16 \\ &= 0.5 + 0.5 + 1.0 \\ &= 2 \text{ bits/symbol} \end{aligned}$$



Encoding the letters

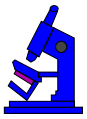
A fixed size 3-bit code, and then a more complex code:

Symbol	3-bit code	Complex code
A	000	0
B	001	10
C	010	1100
D	011	1101
E	100	1110
F	101	1111

Analysis of encoding

The average length of the binary digits needed to encode a typical sequence of symbols using the 3-bit code is

$$
\begin{aligned}
L(X) &= \sum_{i=1}^{n} P_{x_i} \bullet \text{sizeof}(x_i) \\
&= \frac{1}{2} * 3 + \frac{1}{4} * 3 + \frac{4}{16} * 3 \\
&= 1.5 + 0.75 + 0.75 \\
&= 3 \text{ bits/symbol}
\end{aligned}
$$



Analysis of encoding

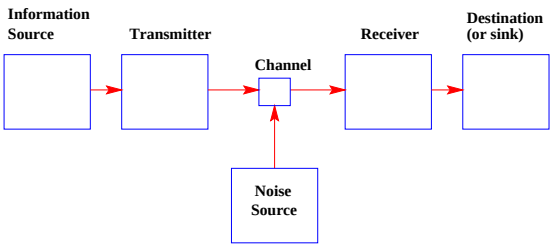
The average length of the binary digits needed to encode a typical sequence of symbols using the complex encoding is

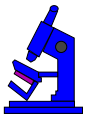
$$\begin{aligned} L(X) &= \sum_{i=1}^{n} P_{x_i} \bullet \text{sizeof}(x_i) \\ &= \frac{1}{2} * 1 + \frac{1}{4} * 2 + \frac{4}{16} * 4 \\ &= 0.5 + 0.5 + 1.0 \\ &= 2 \text{ bits/symbol} \end{aligned}$$

i.e. it is more efficient, averaging only 2 bits for each symbol transmitted.

Entropy and transmission rate

If our source was transmitting 0 and 1 bits with equal probability, but the received data was corrupted 50% of the time, we might reason that our rate $r(X)$ of information transmission was 0.5, because half of our data is getting through correctly.





Entropy and transmission rate

However, a better argument is to consider the difference between the entropy of the source and the conditional entropy of the received data:

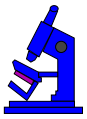
$$r(X) = H(X) - H(X \mid y)$$

where $H(X \mid y)$ is the *conditional* entropy of the received data.

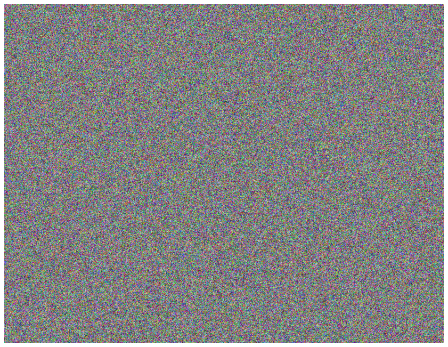
Entropy and transmission rate

$$\begin{aligned} H(X \mid y) &= 0.5 * \log_2 2 + 0.5 * \log_2 2 \\ &= 1 \\ \text{and} \quad H(X) &= 1 \qquad \text{(shown before)} \\ \text{so} \quad r(X) &= H(X) - H(X \mid y) \\ &= 0 \ \text{bits/symbol} \end{aligned}$$

This is a much better measure of the amount of information transmitted.



Example



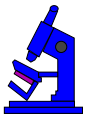
The right hand image has been encoded as a stream of bits, and then each bit has been selected and - on the flip of a coin - changed or not. As you can see - no information is present.

Huffman encoding

An immediate question of interest is "*What is the minimum length bit string that may be used to compress a string of symbols?*".

The Huffman encoding minimizes the bit length given the frequency of occurence of each symbol[3]. The resultant bit string in the best case will be the length predicted from the calculation of the source entropy.

[3]Note that it presupposes knowledge about these frequencies.

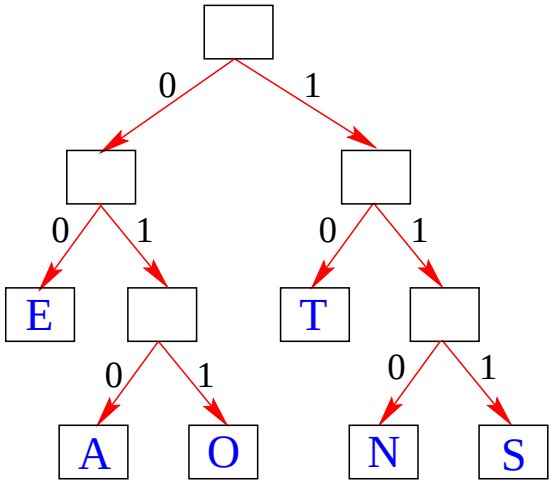


Huffman encoding

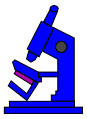
✔ How can we get knowledge about the frequency of (say) the letters in the English language?

(answer)

Huffman encoding



Less common characters use longer bit strings.



Huffman encoding

Our algorithm for encoding is simple - we calculate the tree encoding knowing the frequency of each letter:

Symbol	Coding
E	00
T	10
A	010
O	011
N	110
S	111

To decode, traverse the tree taking a left or right path according to the bit. The leaf has our symbol.

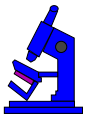
Redundancy

The ratio of the entropy of a source $H(X)$ to what it would be if the symbols had equal probabilities $H'(X)$, is called the *relative* entropy. We use the notation $H_r(X)$, and

$$H_r(X) = \frac{H(X)}{H'(X)}$$

The *redundancy* of the source is $1 - H_r(X)$

$$R(X) = 1 - H_r(X)$$



Redundancy

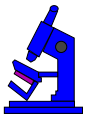
- ✔ If we look at English text a symbol at a time[4], the redundancy is about 0.5.
- ✔ This indicates that it should be simple to compress English text by about 50%.
- ✔ This sort of redundancy is a *unitless* relative redundancy

[4] That is, without considering letter *sequences*.

Unicity distance

Defined by Shannon - an approximation to the amount of ciphertext such that the the sum of the source entropy and the encryption key entropy is the same as the number of ciphertext bits used.

- ✔ Ciphertexts longer have only one meaningful decryption
- ✔ Ciphertexts shorter may have more than one meaningful decryption (and hence be stronger, as a hacker will not know which one is correct)



Unicity distance

- ✔ The longer the unicity distance, the better the cryptosystem
- ✔ Unicity distance U is the entropy of the key divided by the redundancy of the source, and is approximately

$$U \approx \frac{\log_2 K}{R \log_2 P}$$

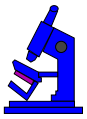
(K key size, R redundancy, P number of symbols).

Unicity distance

26 letter alphabet, and 26! keys

$$\begin{aligned} U &\approx \frac{\log_2 26!}{0.5 \log_2 26} \\ &\approx \frac{88}{0.7 * 4.7} \\ &\approx 27 \end{aligned}$$

So given a ciphertext of 27 symbols, a unique decoding is possible.



Unicity distance

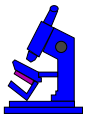
In general

- ✔ Longer key length then longer unicity distance
- ✔ Redundancy inversely proportional to unicity distance
- ✔ Estimates the minimum amount of ciphertext for which there is only a single plaintext solution on doing a brute force attack...

Information flow

- ✔ We may also more abstractly model some security policies by considering the flow of information in a system.
- ✔ We can use *entropy* to formalize this.
- ✔ In this context, we can establish quantitative results about information flow in a system, rather than just making absolute assertions[5].

[5] For example, "*System X reveals no more than 25% of the input values*".



Information flow

We have a definition of information flow based on the conditional entropy $H(x \mid y)$ of some x given y:

Definition 5. The command sequence c causes a flow of information from x to y' if $H(x \mid y') < H(x \mid y)$. If y and x are independant, then $H(x \mid y) = H(x)$.

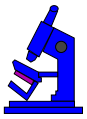
We can use this to detect *implicit* flows of information, not just explicit ones in which we directly modify an object.

Information flow

Consider this example:

```
if x>5 then
   y := 0
else
   y := 1;
```

After this code segment, we can determine if $x > 5$ from y' even though we do not ever assign y' directly from some function of x. In other words we have an implicit flow of information from x to y'.



Information flow

Formal treatment by considering the entropy of x. If the likelihood of $x > 5$ is 0.5, then $H(x) = 1$. We can also deduce that $H(x \mid y') = 0$, and so

$$H(x \mid y') < H(x \mid y) = H(x) = 1$$

and information is flowing from x to y'. Paper gives some background.