Information Processing and the Brain

Josh Felmeden

October 5, 2021

Contents

1 Information Theory

Information theory quantifies the amount of information that is potentially available from the communication channel using. So, we look to answer the question: can the receiver decide how informative the channel is likely to be; how much information is in the channel.

1.1 Randomness

The key is that if the information in the channel is predictable, the receiver is not going to learn much from this. Therefore, we will look at randomness (coin flipping, geiger counters, etc).

We will also look and unexpectedness, and consider the relationship between randomness, unexpectedness, and information theory.

Netflix is a good example of this. Their star rating of films helps them to recommend films to you, and use your experiences to recommend films to others. The recommended tab is the *channel* in the information theory. The films are the medium, and Netflix is communicating to the user information about the film. If Netflix recommends a really popular film, the information is predictable, and the user is not learning much. Therefore, we are not gaining much information about the quality of the films.

Another example is the star rating of movies. The star rating is not 'random' enough for the consumer, in that there is not enough information that we as the user can learn from this. A good film with a lot of hype will likely have a good star rating, and therefore we would already know this.

Therefore, we can say that a statement, despite being correct, can be useless if we can predict it.

> *The theory of information starts with an attempt to allow us to quantify the informativeness of information, but not its salience or validity.*

2 Shannon's entropy

For a finite discrete distribution with a random variable X, possible outcomes $\{x_1, , x_2, \ldots, x_n\} \exists \mathcal{X}$ and a probability mass function p_x giving probabilities $p_X(x_i)$, the entropy is:

$$H(X) = -\sum_{x_i \exists \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

This formula *quantifies* some information received. In the example of star ratings on netflix:

1 star	0.016
2 star	0.310
3 star	0.627
4 star	0.057

We obtain the following formula:

$$H(X) = -0.016 \log_2 0.016 - 0.31 \log_2 0.31 - 0.627 \log_2 0.627 - 0.057 \log_2 0.057 \approx 1.28$$

Choosing base two is arbitrary, but because we are dealing with information, it is useful to deal in bits. In other circumstances, it is possible to choose a different base. The result, 1.28 is the entropy for the channel of Netflix film star ratings.

If the star ratings were all equally likely, we would get an entropy of:

$$H(X) = -4 \times 0.25 \log_2 0.25 = 2$$

This is higher value entropy meaning the information is more useful.

In contrast, if all films are rated one star, we end up with entropy 0, meaning that it is useless. We assume $\log 0$ is 0 for shannon's entropy despite this not being the case in reality.

2.1 Nice properties

Shannon's entropy works on any sample space. It's good because you can calculate this on things that would otherwise be impossible, such as items bought from a grocers. This is because it is defined on probability space, while other calculations are calculated in vector space (such as mean, etc).

It's **always positive**, unless it is zero, which is when the distribution *isn't random* (aka *determined*).

If the distribution is *uniform*:

$$p_X(x_i) = \frac{1}{n}$$

for all x_i where (# is equivalent to `len`)

$$n = \#\mathcal{X}$$

then, since $-\log_2(1/n) = \log_2 n$

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

We won't prove it here, but is not difficult to prove that:

$$0 \leq H(X) \leq \log_2 n$$

with $H(X) = 0$ *only* if one probability is one and the rest zero, and $H(X) = \log_2 n$.

2.1.1 Case of n = 2

With two outcomes, a and b with $p(a) = p$ and $p(b) = 1 - p$ then

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

The resulting graph of entropy will be symmetric; rising towards 1.

2.1.2 Source coding

the main reason to believe that Shannon's entropy is a good quantity for calculating entropy is its relationship to so called source coding.

Consider storing a long sequence of letters `A, B, C, D` as binary (AABACBDA for example). If we wanted to digitise this, we might use a very simple binary representation of these characters (`A = 00, B = 01, ...`). Using this representation means the average bits per letter is 2. Now, if we had different distribution of letters, then we could possibly come up with a smaller, more compact version of the code. For example, if we had the following distribution:

A	B	C	D
0.5	0.25	0.125	0.125

We could use the following encoding:

A	B	C	D
0	10	110	111

Don't forget that this resulting code should be *prefix free*.

Now, we can prove that this code is shorter on average:

$$L = 0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75 < 2$$

3 Types

Term e is **well-typed** iff there is τ such that $\emptyset \vdash e : \tau$ is derivable according the the *static* rules of the language.

Say we want to prove the following:

$$\emptyset \vdash \ \mathsf{let}(\mathsf{str}[\mathsf{my}]; x, (\mathsf{times}(\mathsf{len}(x); \mathsf{num}(0))))$$

Type systems restrict the set of allowed programs.

3.1 Basic properties of typing

Lemma (Inversion of Typing): Suppose that $\Gamma \vdash e : \tau$. If $e = \text{plus}(e_1; e_2)$ then $\tau = \text{num}, \Gamma \vdash e_1 :$ num and $\Gamma \vdash e_2 :$ num and similarly for the other constructs of the language.

Lemma (Unicity of Typing): For every typing context Γ and expression e there exists at most one τ such that Γ such that $\Gamma \vdash e : \tau$.

Lemma (Weakening): If $\Gamma \vdash e' : \tau'$ then $\Gamma, x : \tau \vdash e' : \tau'$ for any $x \notin dom(\Gamma)$ and any type τ.

Lemma (Substitution): If $\Gamma, x : \tau \vdash e' : \tau'$ and $\Gamma \vdash e : \tau$ then $\Gamma \vdash e'[e/x] : \tau'$