0.1 Reading for information theory

Please read this brief introduction to information theory. After reading, please mark out where it is the easiest and most difficult to understand.

0.1.1 Measuring information amount

One is presumably familiar with the computer terminology "bits". For instance, an integer between 0-255 needs 8 bits to represent or convey it, so, the integer 15 is represented by 8 binary digits as 00001111. Before you know anything about that integer, you may know that its can be equally likely any one integer from 0 up to 255, i.e., it has a probability of $P(n) = 1/256$ to be any $n \in 0 - 255$. However, once someone told you the exact number, say $n = 10$, this integer has a probability $P(n) = 1$ for $n = 10$ and $P(n) = 0$ otherwise, and you need no more bits of information to know more about this integer.

Note that $\log_2 256 = 8$, and $\log_2 1 = 0$. That is, before you know which one among the 256 possibilities n is, it has

$$-\log_2 P(n) = \log_2 256 = 8 \text{ bits} \tag{1}$$

of information missing from you. Once you know $n = 10$, you miss no bits of information since $-\log_2 P(n = 10) = 0$.

Similarly, if you flip a coin, and each flip gives head or tail with equal probability, then there is only one bit of information regarding the outcome of the coin flipping, since there is a probability $P = 1/2$ for either head or tail, giving $-\log_2 P(\text{head}) = -\log_2 P(\text{tail}) = 1$ bit. Suppose that you can get information about integer $n \in [0, 255]$ by coin flipping. Say the first coin flip says by head or tail whether $n \in 0 - 127$ or $n \in 128 - 255$. After this coin flip, let us say that it says $n \in 0 - 127$. Then you flip the coin again, and this time to determine whether $n \in 0 - 63$ or $n \in 64 - 127$, and then you flip again to see whether the number is in the first or second 32 integers of either interval, and so on. And you will find that you need exactly 8 coin flips to determine the number exactly. Thus, an integer between 0-255 needs 8 bits of information. Here, one bit of information means an answer to one "yes-no" question, and n bits of information means answers to n "yes-no" questions.

Now let us say that we are flipping a special coin with $P(\text{head}) = 9/10$ and $P(\text{tail}) = 1/10$. So before the coin is even flipped, you can already guess that the outcome is most likely to be "head". So the coin flipping actually tells you less information than you would need if the outcomes were equally likely. For instance, if the outcome is "head", then you would say, well, that is what I guessed, and this little information from the coin flip is almost useless except to confirm your guess, or useful to a smaller extent. If the coin flip gives "tail", it surprises you, and hence this information is more useful. More explicitly,

$$\begin{aligned} -\log_2 P(\text{head}) &= -\log_2 9/10 \approx 0.152 \text{ bit} \\ -\log_2 P(\text{tail}) &= -\log_2 1/10 \approx 3.3219 \text{ bit} \end{aligned}$$

So, a outcome of "head" gives you only 0.152 bit of information, but a "tail" gives 3.3219 bits. If you do many coin flips, on average each flip gives you

$$P(\text{head})(-\log_2 P(\text{head})) + P(\text{tail})(-\log_2 P(\text{tail})) = 0.9 \cdot 0.152 + 0.1 \cdot 3.3219 = 0.469 \text{ bit} \tag{2}$$

of information, less than the one bit of information if head and tail are are equally likely. More generally, the average amount of information for probability distribution $P(n)$ for variable n is

$$I = -\sum_n P(n) \log_2 P(n) \text{ bits} \tag{3}$$

The formula for information is the same as that for entropy, which we denote by $H(n)$ as the entropy on variable n. When signals are represented as discrete quantities, we often use entropy H and information I inter-changably to mean the same thing.

In the coin flip example, the amount of information is higher when the head and tail occur equally probable (1 bit), than when it is not (e.g., 0.469 bit). In general, the amount of entropy or information on variable n is more when the distribution $P(n)$ is more evenly distributed, and most in amount when $P(n) =$ constant, i.e., exactly evenly distributed. So if variable n can take N possibilities, the most amount of information is $I = \log_2 N$ bits, hence 8 bits for an integer $n \in [0, 256]$. Hence, a more evenly distributed $P(n)$ means more varibility in n, or more randomness, or more ignorance about n before one knows its exact value.

0.1.2 Information transmission and information channels

Let a signal S be transmitted via some channel to a destination giving output O. The channel can have some noise, and let us assume

$$O = S + N \tag{4}$$

So for instance, S can be the input at the sensory receptor, and O can be the output when it is received at a destination neuron. Before you receive O, all you have is the expectation that S has a probability distribution $P_S(S)$. So you have

$$H(S) = -\sum_S P_S(S) \log_2 P_S(S) \ \text{ bits} \tag{5}$$

of ignorance or missing information about S. Let us say that you also know the channel well enough to know the probability distribution $P_N(N)$ for the noise N. Then you receive a signal O, and you can have a better guess on S, as following a probability distribution $P(S|O)$, which is the conditional probability of S given O. As you can imagine, $P(S|O)$ must have a narrower distribution than $P_S(S)$. For instance, if you know originally that S is between -10 to 10, and you know that the noise is mostly $N \in -1, 1$, and if you received an $O = 5$, then you can guess that $S \in (4, 6)$. So your guess on S has narrowed down from $(-10, 10)$ to $(4, 6)$. If S can only take on the 21 integer values (for instance), you originally had about $\log_2 21 = 4.4$ bits of information missing. Now given O, you have only about $\log_2 3 = 1.59$ bits of information missing from you. So given output O, you can guess what S is to some extent, though not as good as if you received S directly. The amount of information still missing is the conditional entropy

$$\begin{aligned} H(S|O) &\equiv -\sum_S P(S|O) \log_2 P(S|O) && (6) \\ &= 1.59 \text{ bits in the example above} && (7) \end{aligned}$$

which should be much small than $-\sum_S P_S(S) \log_2 P_S(S)$ which in the above example is 4.4 bits. So the amount of information O tells you about S is then, for this particular value of output O,

$$\begin{aligned} H(S) - H(S|O) &= [-\sum_S P_S(S) \log_2 P_S(S)] - [-\sum_S P(S|O) \log_2 P(S|O)] && (8) \\ &= 4.4 - 1.59 = 2.8 \ \text{ bits, in the example above} \end{aligned}$$

The first and second terms are the amount of information missing about S before and after, respectively, knowing O.

Each input S gives a conditional probability distribution $P(O|S)$ (which is probability of O given S) of the output O. Assuming that the noise N is independent of S, we know that $O = S + N$

should differ from S by an amount dictated by the noise which follows a probability $P_N(N)$, hence $P(O|S) = P_N(O - S)$, i.e., the probability that O occurs given S is equal to the probability $P_N(N = O - S)$ that the noise value $N = O - S$ occures. In different trials, you will receive many different output signals O, arising from randomly drawn inputs S from its probability distribution $P(S)$. Hence, over all trials, the overall probability distribution of $P_O(O)$, which is called the marginal distribution, can be obtained by weighted summation of the conditional probability $P(O|S)$ by its occurrance weight $P(S)$, i.e.,

$$P_O(O) = \sum_S P_S(S) P(O|S) = \sum_S P_S(S) P_N(O - S). \tag{9}$$

So, when averaged over all outputs O, the information that O contains about S is obtained simply by averaging the quantity in equation (8) by probability $P_O(O)$, as

$$\begin{aligned} I(O;S) &\equiv H(S) - \sum_O P_O(O) H(S|O) \\ &= [-\sum_S P_S(S)\log_2 P_S(S)] - [-\sum_{O,S} P(O)P(O|S)\log_2 P(S|O)] \\ &= [-\sum_S P_S(S)\log_2 P_S(S)] - [-\sum_{O,S} P(O,S)\log_2 P(S|O)] \end{aligned}$$

Here $P(O, S) = P_O(O)P(S|O)$ is the joint probability distribution of O and S. If an information channel transmits $I(O; S)$ bits of information from source S to output O per unit time, then this channel is said to have a capacity of at least $I(O; S)$ bits per unit time.

A particular useful example is when S and N are both gaussian,

$$P(S) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-S^2/(2\sigma_s^2)} \qquad P(N) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-N^2/(2\sigma_n^2)} \tag{10}$$

with zero means and variances σ_s^2 and σ_n^2 respectively. Then,

$$\begin{aligned} P_O(O) &= \int dS P(S) P(O|S) \propto \int dS e^{-S^2/(2\sigma_s^2)} e^{-(O-S)^2/(2\sigma_n^2)} & (11) \\ &= \frac{1}{\sqrt{2\pi(\sigma_s^2 + \sigma_n^2)}} e^{-O^2/(2(\sigma_s^2+\sigma_n^2))} & (12) \\ &\equiv \frac{1}{\sqrt{2\pi}\sigma_o} e^{-O^2/(2\sigma_o^2)} & (13) \end{aligned}$$

Hence, O is also a gaussian random variable, with zero mean and variance $\sigma_s^2 + \sigma_n^2$. It can be shown that entropy of gaussian signals is, within a constant, the log of standard deviation of the signal. For example, $H(S) = \int dS P(S) \log_2 P(S) = \log_2 \sigma_s$ + constant. Then, the amount of information in O about S is

$$\begin{aligned} I(O;S) &= H(S) - H(S|O) & (14) \\ &= H(O) - H(O|S) = H(O) - H(N) & (15) \\ &= \frac{1}{2}\log_2(1 + \frac{\sigma_s^2}{\sigma_n^2}) = \log_2 \frac{\sigma_o}{\sigma_n}, & (16) \end{aligned}$$

which depends on the signal-to-noise ratio σ_s^2/σ_n^2. Note that we have equated $H(N) = H(O|S)$, and $H(S) - H(S|O) = H(O) - H(O|S)$ (see below).

Equation (16) gives an intuitive understanding of the mutual information $I(O; S)$ for gaussian signals. Imagine an output signal O which can vary within a range σ_o, and we discretize it into

$\frac{\sigma_o}{\sigma_n}$ values, with quantization step size σ_n determined by the size of the noise. When each of the $\frac{\sigma_o}{\sigma_n}$ discrete values is equally likely to occur, the information provided by each discrete value is $\log_2 \frac{\sigma_o}{\sigma_n} = I(O;S)$ — naturally, as O, having a range of σ_o, conveys information about S with a resolution of σ_n.

0.1.3 Mutual information, information redundancy, and error correction

One can say that once O is known, one knows something about S. This means O and S share some information, whose amount is exactly $I(O;S)$, which is called mutual information. The difference between O and S is caused by noise, and that is the information not shared between S and O. Hence, this mutual information is symmetric between O and S,

$$I(O;S) = I(S;O) = \sum_{O,S} P(O,S) \log_2 \frac{P(O,S)}{P(S)P(O)} \tag{17}$$

This symmetry is the reason why we equated $H(S) - H(S|O) = H(O) - H(O|S)$ in equation (15).

We can use this result in another situation where information is shared between nearby pixels in images. Let S_1 and S_2 be the image intensities in two nearby pixels of an image. Normally, these two intensities are likely to be similar in most natural images. Hence, if you know S_1, you can already guess something about S_2. Or, $P(S_2|S_1) \neq P(S_2)$, so S_1 and S_2 are not independent variables. $P(S_2|S_1)$ usually has a narrower distribution than $P(S_2)$. So we say that information provided by S_1 and S_2 are somewhat redundant, although information provided by S_2 is not exactly the same as that by S_1. When there is information redundancy, we have $H(S_1) + H(S_2) > H(S_1, S_2)$, i.e., the summation of the amount of information provided by S_1 and S_2 separately is larger than the information contained by the two signals together. Then the amount of mutual information between S_1 and S_2 is

$$I(S_1; S_2) = H(S_1) + H(S_2) - H(S_1, S_2) \tag{18}$$

In general, given N signals S_1, S_2, ..., S_N,

$$\sum_i H(S_i) \geq H(S_1, S_2, ..., S_N) \tag{19}$$

with equality when all S's are independent or when there is no redundancy. One may quantify the degree of redundancy by

$$\text{Redundancy} = 1 - H(S_1, S_2, ..., S_N)/[\sum_{i=1}^{N} H(S_i)] \tag{20}$$

which takes a value between 0 and 1, with 0 meaning no redundancy and 1 meaning complete redundancy (which will not occur in reality unless $N \to \infty$ and $S_i = S_1$).

Redundancy exists in many natural information representation such as natural images or natural languages (represented as a string of letters, and the nearby letters are correlated). When information is represented redundantly, we say that the representation is not efficient. In our example, if $\sum_i H(S_i)$ = 100 bits $> H(S_1, S_2, ..., S_N)$ = 50 bits, it is not efficient to use 100 bits to represent 50 bits of information. Sending the signals $\mathbf{S} \equiv (S_1, S_2, ..., S_N)$ (per unit time) through an information channel in this form would require a channel capacity of at least 100 bits per unit time. Shannon and Weaver (1949) showed that theoretically, all the information (of amount $H(S_1, S_2, ..., S_N)$) about $\mathbf{S} \equiv (S_1, S_2, ..., S_N)$ could be faithfully transmitted through a channel of a capacity of only $H(S_1, S_2, ..., S_N)$ (e.g., 50 bits) per unit time, by encoding $\mathbf{S}$ into some other form $\mathbf{S}' = f(\mathbf{S})$, where

$f(.)$ is an (invertable) encoding transform. In such a case, $\mathbf{S}'$ would be a more efficient representation of the original information in $\mathbf{S}$, and the information channel would be more efficiently used.

Redundancy is useful for the purpose of error correction. In other words, while efficient coding or representation of signals may save information storage space or information channel capacity, it also reduces or removes the ability to recover information in the face of error. For instance, given a sentence conveyed noisily as "I lik. .o invite y.u f.r din.er" (in which each "." indicates some missing letter(s)), one can recover the actual sentence "I like to invite you for dinner" using the knowledge of the structures in the natural language. This structure in a natural language is caused by the redundancy of information representation, so that one can predict or guess some signals (letters) from other signals (letters), i.e., there are non-zero mutual information between different letters or words in a sentence or sentences. In terms of probability and information, this can be stated as follows. Without any neighbooring letters or context, one can guess a missing letter S as one of any 26 letters in the alphabet with probability $P(S)$ (though some are more likely than others), and one would require an information amount $H(S) = -\sum_S P(S)\log_2 P(S)$ to obtain this letter; With the neigboring letters, the redundancy between the letters enables the guess to be narrowed down to fewer choices, i.e., the conditional probability $P(S|\text{contextual letters })$ has a narrower distribution over the 26 letters in the alphabet, so that the amount of information needed to recover the letter is the conditional entropy $H(S|\text{contextual letters})$, which is less than $H(S)$ given the redundancy. Redundancy in natural languages enable us to communicate effectively through noisy telephone lines, or when one speaks with imperfect grammar or unfamiliar accent. If everybody speaks clearly with standard accent and perfect grammer, redundancy in language would be less necessary. How much redundancy is optimal in a representation depends on the level of noise, or tendency to errors, in the system, as well as the end purpose or task that utilizes the transmitted information.