Example: What is the probability that the sum of two fair dice is at least 7, given that the product is 6?

Let A be the event that the sum is at least 7, and let B the event that the product is 6. By definition, the conditional probability is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

By trial and error, the possible ordered pairs of values giving a product of 6 are $(1,6)$, $(2,3)$, $(3,2)$, and $(6,1)$ (ordered by size of the first in the pair). Testing each one in the list, only the first and last have a sum at least 7. Thus, since these pairs are distinct atomic events each of probability $1/(6 \cdot 6) = 1/36$,

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$= \frac{P(1,6) + P(6,1)}{P(1,6) + P(2,3) + P(3,2) + P(4,6)}$$

$$= \frac{1/36 + 1/36}{1/36 + 1/36 + 1/36 + 1/36} = \frac{1}{2}$$

Entropy, uncertainty

The **entropy** of a situation is roughly the *uncertainty* in it. It is not clear from this vague idea what the best way would be to give a more quantitative definition, but Hartley and Shannon figured this out (see below).

For example, the outcome of a *fair* coin is *more uncertain* than that of a *biased* coin which gives heads 2/3 of the time.

The outcome of a single fair die is more uncertain than flipping a coin: there are more things that can happen, each with small probability.

We can skim newspapers and other 'lightweight' text since most of the words are not vital to the content. If we ignore many of them the message still comes through: the information content is low, and information is repeated. By contrast, technical material is harder to read because it is more succinct, preventing us from skip over things. It is not as repetitive. Equivalently, there is a high **information rate**.

In ordinary language refering to ordinary events, in the sentence

`In January I had to shovel` <*blank*>

the missing word does not leave much uncertainty. Most likely it was `snow`, though with much lower probability it could have been something else. Filling in the word does not *add* much info. By contrast

`My old car is` <*blank*>`.`

contains a great deal of uncertainty, since there is a huge range of fairly likely possibilities: `reliable, lousy, cheap, ugly`, etc. That is, filling in the word *adds considerable information.*

In lower-level structure of language, most typographical errors in ordinary text are not hard to correct, because of the **redundancy** of natural languages such as English. For example,

`The snow was clean and wht.`

is easy to correct to

`The snow was clean and white.`

On the other hand, typographical errors are harder to detect and correct in technical writing. For example, is the following correct, or not?

$$\frac{1}{1^4} + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \frac{1}{5^4} + \ldots = \frac{\pi^4}{90}$$

The idea of **compression** (also called *source coding*) is that something with a lot of redundancy can be made smaller without loss of information.

The idea of **error-correction** (also called *forward error correction* to distinguish it from check-bits and such) is that redundancy can be *added* to things in a clever way to make it more robust against damage.

We will only talk about a low-level *syntactical* version of **information** and **entropy**, since this can be formalized. More interesting but subtler questions about *semantic* information are too sophisticated for this context.

That is, we will *abstract* the notion of *information* away from the colloquial notion of information.

The **self-information** of an event A is

$$I(A) = -\log_2 P(A)$$

An unlikely event has greater (self-) information than a relatively likely event.

For a fair coin the sample space is $\{\texttt{heads}, \texttt{tails}\}$ and

$$P(\texttt{heads}) = P(\texttt{tails}) = \frac{1}{2}$$

The self-information of either head or tail is

$$I(\texttt{H}) = -\log_2 \frac{1}{2} = 1 \quad I(\texttt{T}) = -\log_2 \frac{1}{2} = 1$$

This motivates the name for the **unit** of information, the **bit**.

The **entropy** $H(\Omega)$ of a sample space Ω is the *expected value* of the self-information of atomic events in Ω:

$$\text{entropy } = H(\Omega) = \sum_{1 \le i \le n} P(\omega_i)\, I(\omega_i)$$

$$= \sum_{1 \le i \le n} -P(\omega_i) \log_2 P(\omega_i)$$

Thinking of sample spaces as experiments or tests, entropy is a measure of information acquired by doing the experiment, or how much uncertainty is eliminated.

Since we only care about the probabilities $p_1, \ldots, p_n$, we can suppress reference to the sample space and just refer to the probabilities

$$H(p_1, \ldots, p_n) = \text{ entropy of sample space}$$

$$\{\omega_1, \ldots, \omega_n\} \text{ with } P(\omega_i) = p_i$$

Define the entropy of a random variable similarly. For X a random variable on sample space $\Omega = \{\omega_1, \ldots, \omega_n\}$ entropy of X is an expected value

$$H(X) = \sum_{\text{values } x} -P(X = x) \log_2 P(X = x)$$

Theorem: Any entropy function $H(p_1, \ldots, p_n)$ meeting the conditions below is a positive scalar multiple of the entropy function we just defined:

- $H(p_1, \ldots, p_n)$ is maximum when $p_1 = \ldots = p_n = \frac{1}{n}$. That is, the most pre-existing uncertainty is when all possibilities are equally likely.

- For any permutation $i \to s(i)$ of the indices,

$$H(p_1, \ldots, p_n) = H\left(p_{s(1)}, \ldots, p_{s(n)}\right)$$

That is, only the probabilities matter, not their ordering or labeling.

- $H(p_1, \ldots, p_n) \geq 0$, and is 0 only if one of the p_is is 1. That is, uncertainty disappears entirely only if there is no randomness present.

- $H(p_1, \ldots, p_n) = H(p_1, \ldots, p_n, 0)$. That is, 'impossible' outcomes do not contribute to uncertainty.

•
$$H(\underbrace{\frac{1}{n},\ldots,\frac{1}{n}}_{n}) \leq H(\underbrace{\frac{1}{n+1},\ldots,\frac{1}{n+1}}_{n+1})$$

That is, a larger ensemble of equally likely possibilities is more uncertain than a smaller ensemble.

• H should be a continuous function of the probabilities: 'small' changes in the probabilities should not cause 'large' changes in uncertainty.

• For positive integers m, n,

$$H\left(\frac{1}{mn},\ldots,\frac{1}{mn}\right)$$

$$= H\left(\frac{1}{m},\ldots,\frac{1}{m}\right) + H\left(\frac{1}{n},\ldots,\frac{1}{n}\right)$$

That is, the uncertainty in performing two independent experiments should be the **sum** of the two uncertainties.

• Let $p = p_1 + \ldots + p_m$ and $q = q_1 + \ldots + q_n$ with all p_i and q_j positive, and $p + q = 1$. Then

$$H(p_1, \ldots, p_m, q_1, \ldots, q_n)$$

$$= H(p, q) + pH(p_1/p, \ldots, p_m/p) + qH(q_1/q, \ldots, q_n/q)$$

This is about *conditional probabilities* and a sensible requirement about uncertainty in such a situation. That is, we *group* the outcomes of an experiment into two subsets and then say that the uncertainty is the uncertainty of which batch the outcome falls into, plus the *weighted* sum of the uncertainties about exactly where the outcome falls in the subsets.

The simplest examples:

Example: The entropy in a single flip of a fair coin is

$$H\left(\text{coin}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right)$$

$$= \frac{1}{2}\left(-\log_2 \frac{1}{2}\right) + \frac{1}{2}\left(-\log_2 \frac{1}{2}\right)$$

$$= \frac{1}{2}\left(-(-1)\right) + \left(-(-1)\right) = 1 \text{ bit}$$

If we label the coin '0' and '1' then such a coin toss just determines the value of a *bit*.

Example: The entropy in a single roll of a die is

$$H\left(\text{die}\right) = H\left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$$

$$= -\sum_{i=1}^{6} \frac{1}{6} \log_2 \frac{1}{6} = \log_2 6 \approx 2.58496250072 \text{ bits}$$

Example: To compute the entropy of the sum of two dice, note that there are $6 \cdot 6 = 36$ equally likely possible rolls, and (!) among these just 1 way to get 2, 2 ways to get 3, 3 ways to get 4, 4 ways to get 5, 5 ways to get 6, 6 ways to get 7, 5 ways to get 8, 4 ways to get 9, 3 ways to get 10, 2 ways to get 11, and 1 way to get 12. Thus, the entropy is

$$H\,(\text{sum two dice})$$

$$= H\left(\frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{5}{36}, \frac{4}{36}, \frac{3}{36}, \frac{2}{36}, \frac{1}{36}\right)$$

$$= -\frac{1}{36}\log_2\frac{1}{36} - \frac{2}{36}\log_2\frac{2}{36} - \ldots - \frac{1}{36}\log_2\frac{1}{36}$$

$$\approx 3.27440191929 \text{ bits}$$

The idea of noiseless (source) coding

Noiseless coding or **source coding** makes data smaller (for transmission or storage) by cleverly *removing* redundancy.

(This type of coding is *not* about noise or errors.)

An important example of noiseless coding is **compression**. Also abbreviations, shorthand, and symbols are important examples.

For simplicity we talk about **lossless** compression, where the original can be perfectly reproduced from the smaller/compressed version. This *includes* GIFs, but *excludes* JPEGs, which are *lossy*.

The notion of **channel** is abstract. Your notebook is a channel. Your brain is a channel. Your backpack is a channel. Also, naturally, wires are channels. But wireless communication is by channels, too.

A **memoryless source** emitting **source words** in a set W emits those words with prescribed probabilities, and the probability that a given word is emitted does *not* depend upon what came before it.

That is, a **memoryless source** is a sequence $X_1, X_2, \ldots$ of independent and identically distributed (**i.i.d.**) random variables on a probability space, taking values in a set W of **source words**.

Example: A simple source emits a stream of 0's and 1's with equal probabilities. Each random variable $X_1, X_2, \ldots$ has **distribution**

$$P(X_i = 0) = \frac{1}{2}$$

$$P(X_i = 1) = \frac{1}{2}$$

Example: A very simple but useful model of English is a memoryless source $X_1, X_2, \ldots$ where the possible values of the random variables X_i are characters a–z, where each character occurs with its average frequency:

$$\begin{aligned} P(X = \text{`e'}) &= 0.11 \\ P(X = \text{`t'}) &= 0.09 \\ &\ldots \end{aligned}$$

Remark: A useful more general type of source is a *Markov source*, where that there is a fixed t so that the n^{th} word emitted depends probabilistically on only the t previous values. Things are complicated enough already for the simpler i.i.d. model.
