Information Theory in Ten Minutes

Spencer Tipping

November 24, 2013

Disclaimer: I'm not an academic, nor have I taken any classes on information theory. I'm a dropout and Wikipedia student who uses it in potentially inappropriate ways at my day job, so anything presented here may be arbitrarily erroneous or misleading. (On the bright side, this guide is probably correct enough that it may make it easier to understand stuff.)

1 Randomness

Like *monad* and *vector space*, *random variable* describes an interpretation of an object, not any intrinsic quality. So a coin is a random variable in the sense that it generates values that can't be predicted. From your perspective, I'm a random variable if we're on the phone and I'm flipping a coin and I'm saying heads or tails each time. If you ask a mathematician why they don't know the outcomes of coin tosses, they'll say the outcomes are random – but really, this is just a mathematically polite excuse for, "we don't want to go to the trouble to predict them."

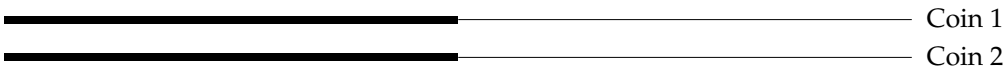
So instead of predictions, a mathematician will give you a distribution of outcomes. This distribution characterizes the relative frequency of the different kinds of observations you might make; so a fair coin can be described as producing heads 50% of the time, and tails the other 50%.

Some random variables are continuous, but information theory doesn't conventionally deal with those, so I'm going to pretend they don't exist for this guide. Since a coin toss is discrete, let's represent the space of outcomes this way (line thickness is just to tell the outcomes apart; it doesn't have any statistical meaning):

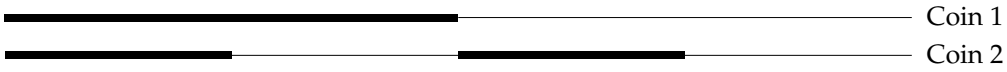
Because heads and tails occupy the same horizontal area, they are equally probable. Here's what a 75%-heads unfair coin looks like:

2 Independence

The diagrams above show how a single coin toss maps into the space of possible outcomes. But the space of possibilities has infinitely many points, so it can describe arbitrarily many coin tosses. Let's map out the outcomes of two fair coin tosses:



There is something very wrong with this picture. It's a valid probability space, but notice that any vertical line we draw through it results in both coins having landed the same way. The only way we'd end up with this kind of probability space is if the two coins were stuck together. If we want them to be separate, we need to rearrange the outcomes like this:



With this new layout, the first coin's outcome has no impact on the behavior of the second coin; even if we know the first one came up heads, the second coin still has the same odds that it originally did. Because the outcomes are arranged this way, we say that the two outcomes are statistically independent; in general, this guide assumes that all random variables generate independent values unless mentioned otherwise.

3 Entropy

Entropy is not a first-date kind of word. Even the seemingly innocuous *information content* will get you funny looks; using either one obligates you to start measuring stuff in fractional quantities of bits,[1] which is not only obscure, but also very boring small talk.

Romantic conversation aside, however, entropy is awesome because it lets you very precisely bound the number of bits you need to describe each new outcome from a random variable (in the long run). Specifically:

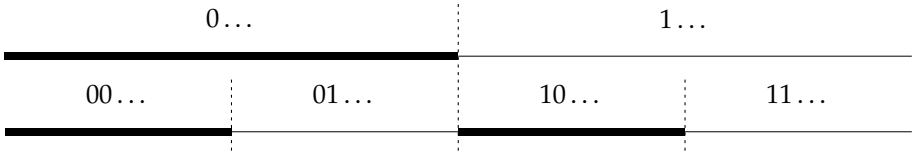
$$\begin{aligned}\text{entropy}(X) &= \sum_{x \in X} -P(x) \cdot \log_2 P(x) \\ &= \text{expected}\left[\frac{\text{bits}}{\text{observation}}\right]\end{aligned}$$

Okay, so how can you possibly have a fractional number of bits and have that be meaningful? Well, remember that entropy is an *expected value*, not a

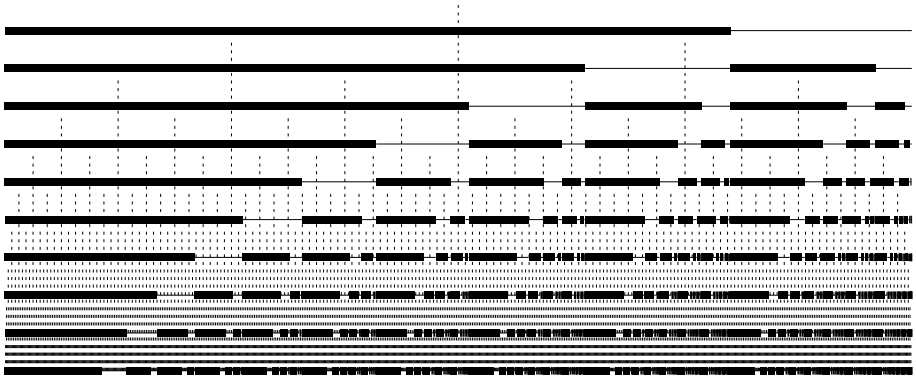
[1] It isn't always measured in bits, actually. Sometimes people change the base of the log to e or 10 to measure in *nats* or *bans*, respectively. But bits are the easiest to work with, especially for programmers, so that's how this guide is written.

single-observation quantity. This means that it's an asymptotic limit of sorts; technically it's the lowest number of bits you could consistently use to describe each new outcome, given convergence to the random variable's probability distribution.[2]

Let's back up for a minute here and ask what a bit really does for us. Beyond being a 0 or a 1, we can think of bits as directions to bisect the probability space. A 0 means that the set of outcomes is on the left-hand side, and a 1 means it's on the right. Trivially, then, we can map each toss of a fair coin onto one bit:



Now watch what happens when the coin is unfair. For example's sake, let's make the coin biased towards heads 80% of the time. By the official entropy formula, we expect each observation to require about 0.72 bits – but let's see what that looks like when we binary-split the probability space at the same rate of one split per observation:



Notice how quickly the vertical binary-split lines converge – much more rapidly than the thick black lines on the left, for example. This means that any outcome consisting of a bunch of heads will require less than one split per observation. In this particular case, once you've split to the leftmost 1/4 of the probability space (which requires two bits), you know the outcomes of the first five coin tosses. So these particular observations are encoded at a rate of under 0.4 bits each. The entropy formula describes the limit case: across every possibility, what is the average ratio of bits to observations?

[2]This convergence is guaranteed by the law of averages, for sufficiently many samples.

3.1 Dismantling the formula

I found the entropy calculation formula to be really confusing the first time I tried to learn information theory. It took a lot of time thinking about it before it started making intuitive sense, but luckily it does if you think about it the right way.

$$\begin{aligned}\sum_{x \in X} -P(x) \cdot \log_2 P(x) &= \sum P(x) \cdot \log_2 \frac{1}{P(x)} \\ &= \sum (\text{fraction of all outcomes}) \times (\text{bits required to split to it})\end{aligned}$$

This is still a little unintuitive, though; why does it suffice to split to a quantity repeatedly given that the slices are so jumbled up in the probability space? Also, does this formula take into account the kinds of "free observations" that you get when the entropy per observation is less than one?

The answer to the first question lies in the fact that the observations are independent. Suppose, for example, that you split exactly down to the subset of space where the first coin landed heads. Then you're right back to where you started, since the first observation's outcome has no impact on any other.[3] So when you look at it this way, encoding an observation is more about reducing the size of the probability space than it is about getting the split lines to line up exactly.

[3] You can see this in the diagram by observing that the area underneath each black bar looks just like the graph as a whole, modulo the split lines.