A Short, Simple Introduction to Information Theory

moultano.wordpress.com/2010/10/23/a-short-simple-introduction-to-3kbzhsxyg4467-7/

Ryan Moulton October 23, 2010

Intuition

Suppose someone tells you that the sun will come up tomorrow. Unless you're in the depths of depression, this probably isn't surprising. You already knew the sun will come up tomorrow, so they didn't really give you much information. Now suppose someone tells you that aliens have just landed on earth and have picked you to be the spokesperson for the human race. This is probably pretty surprising, and they've given you a lot of information. You'd be pretty pissed off if nobody told you. The rarer something is, the more you've learned if you discover that it happened.

An Eight-sided Die

Suppose you have an eight-sided die with the sides labelled A B C D E F G H. Suppose you want to record a series of rolls on a computer. How many bits are required to encode each roll? Well, there are 8 possibilities that are all equally likely, and with n bits you can encode 2^n possibilities, so this requires $\log_2 8 = 3$ bits per roll.

Now, suppose someone tells you that they rolled a vowel. How many bits does it take to encode that roll, now that you already know that the roll was a vowel. There are only two possibilities, A and E, so you can store that roll in $\log_2 2 = 1$ bit. They've just saved you 2 bits by giving you some information.

This is the central thing to understand about information theory. There's some randomness whose outcome is encoded in bits. Someone tells you something about an outcome, and as a result you can store it in fewer bits. The difference in the number of bits is the amount of information you learned. In this case, knowing that they rolled a vowel saves you two bits, so that's how much information they gave you.

Now what happens if someone tells you that they didn't roll a vowel? Now there are 6 possible outcomes, which isn't a power of 2, so we can't trivially map it to some number of bits and we have to start doing math.

Optimal Codes and Entropy

Suppose we're encoding rolls of a die with many sides, and each possible roll is no longer equally likely. How many bits should we use to store each roll? Intuitively we'd like to make the common cases shorter and the rarer cases longer, so that on average, the messages are shorter. More formally, let l_i be the length of the string of bits we use to encode outcome i. We would like to minimize the expected value of l, or .

What are the constraints on how short our encodings can be? We certainly can't have

more than two outcomes encoded in only 1 bit, but how can we generalize this? Think of this in terms of the fraction of the "namespace" that each outcome uses. If an outcome is encoded as 1, then you've already used half of the namespace. No other outcome can start its encoding with 1. Suppose you're encoding an outcome as 11. You're now using half-of-the half-of-the namespace. No other outcome can start with 11, but they are free to start with 0 or 10. Formally, this means that if we encode an outcome using l_i bits, it uses $1/2^{l_i}$ of the namespace. We now have the nice constraint that $\sum \frac{1}{2^{l_i}} \leq 1$.

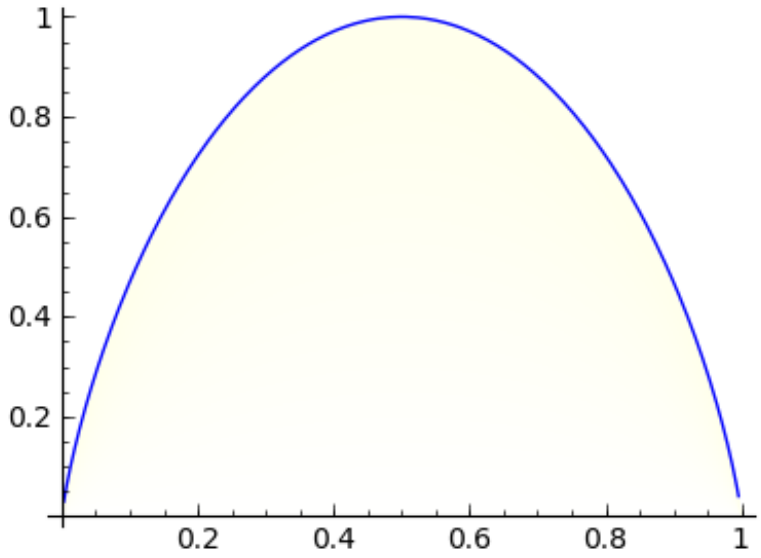
$$\mathrm{E}[l] = \sum p(i) l_i$$

Minimizing $\mathrm{E}[l]$ under this constraint gives[1] you the optimal encoding length for each as $l_i = \log_2 \frac{1}{p(i)}$.

This behaves in the way we would expect. The rarer an outcome, the longer its encoding, and if all of the n outcomes are equally likely, we give each one an encoding of length $\log_2 n$ just like we're used to. Now that we know the optimal length of each encoded outcome, what's the expected encoding length for an event?

We call this quantity the "entropy" of a distribution p. Mathematically it is identical to the quantity of the same name from thermodynamics, and you can think of it as a measure of the "spread" or "disorder" of a probability distribution. It has its peak value when all outcomes are equally likely, and its minimum value when there is only one possible outcome. Careful readers will notice that $p \log(1/p)$ is undefined when $p = 0$. Thankfully though, the limit at 0 is 0. This leads to the interesting result that if there is one outcome that is absolutely certain to happen, you can encode it in 0 bits. There are some things so obvious (certain) that they aren't worth saying at all!

$$\sum p(i) \log_2 \frac{1}{p(i)} = H(p)$$



Entropy of a flip of a biased coin

Back now to our eight-sided die. The entropy of the original die roll is 3, and the entropy of the die roll if we know that the letter rolled is a vowel is 1. What's the entropy of the die roll if we know the letter rolled is not a vowel?

Now we have something whose unit is “bits” but whose value includes fractions of a bit. What can we do with this? After all, if we’re only storing one roll, we still need 3 bits to store 6 possibilities. The trick is that we can use fewer bits if we are storing more rolls at once. There are 2 wasted possibilities in those 3 bits we used for the first roll, and if you’re clever, you can use those to encode some information about the next roll. If we’re clever enough, and storing enough, 2.58… is the lower bound on the number of bits required per roll that you’ll converge to with an optimal compression scheme.

$$\sum_{i=1}^{6}(1/6)\log_2(1/(1/6)) = \log_2 6 = 2.58...$$

So if someone tells us that the roll isn’t a vowel, they’ve given us 3 – 2.58 = 0.415 bits of information. Consider that if they ruled out half of the possibilities, they’d have given us 1 bit. Since they ruled out less than half of the possibilities, it makes sense that they gave us less than one bit.

Information Gain and Conditional Entropy

Suppose now that someone has agreed to tell us whether or not the roll is a vowel, but we don’t know in advance which it will be? What’s the expected value of the information they will give us? The roll will be a vowel 2/8 of the time, and not-a-vowel 6/8 of the time. So, take the expectation of how much information they give us in each case. On average, they will give us 0.8 bits of information about each roll. This quantity we’ve just computed is called the “information gain” of knowing whether the roll is a vowel.

$$2/8 * 2 + 6/8 * 0.415... = 0.8111$$

So more formally, if R is the random variable indicating our roll, we’ve computed $H(R) = 3$. It takes 3 bits to store each roll if we know nothing in advance. We’ve also computed and ; the entropy of the roll given that we know the roll was a vowel, and the entropy of the roll given that we know the roll wasn’t a vowel, respectively. What we’ve computed above is:

$$H(R|V = \text{true}) = 1$$

$$H(R|V = \text{false}) = 2.58$$

This formula is the form of information gain for any two random variables, just sum it over all values of the variable you are conditioning on (in this case, V is either true or false.) Now, lets take the information gain, distribute the probabilities, and rearrange a little bit.

$$P(V = \text{true})[H(R) - H(R|V = \text{true})] + P(V = \text{false})[H(R) - H(R|V = \text{false})]$$

$$H(R) - [P(V = \text{true})H(R|V = \text{true}) + P(V = \text{false})H(R|V = \text{false})]$$

We have the original entropy of R minus the expected value of the entropy of R given that we know the value of V. This second term we call the “conditional entropy” and it is denoted $H(R|V)$. You can think of conditional entropy as the expected number of

bits that are “left” in R once you know V.

Information gain is an extremely useful quantity in machine learning. It tells you how much value your classifier could possibly extract out of using a given feature by itself, and is commonly used for feature selection. Anytime you need to sort anything, sorting by information gain/KL-divergence or G-test score will almost certainly give you great results.

Since I’m sure you want to know more . . .

- Information Theory, Inference, and Learning Algorithms is probably the best textbook on the subject and is available as a free pdf from the author’s website. It’s written well enough to be readable for pleasure.
- A Light Discussion and Derivation of Entropy takes a cute approach to deriving entropy from some very basic and fundamental first principles.