CS 4980/6980: Introduction to Data Science © Spring 2018

Lecture 12: Entropy

Instructor: Daniel L. Pimentel-Alarcón Scribed by: Young Son & Max Hostetter

This is preliminary work and has not been reviewed by instructor. If you have comments about typos, errors, notation inconsistencies, etc., please email the scribers.

12.1 Introduction

Claude Shannon discovered entropy, which is a measure of the unavailable resource in a known environment. It can be used to calculate randomness, in our case with the following formula:

$$H(x) = \sum_x P(x) log_2 \left(\frac{1}{P(x)} \right)$$

Let x represent a random variable distributed according to $P(x)$. The function above defines the Entropy of x.

Recall from a previous lecture, a motivation for studying entropy. We have a table of horses and their probability of winning. We want to be able to represent a specific horse winning in the most efficient manner possible.

Table 12.1: Horse Example

Horse	Horse 1	Horse 2	Horse 3	Horse 4	Horse 5	Horse 6	Horse 7	Horse 8
P(winning)	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
Naive Code	000	001	010	011	100	101	110	111
XX Code	1	01	001	0001	000000	000001	000010	000011

The claim is that the "XX Code" is the most efficient possible encoding for this information. We will use Entropy to verify this. With the information in Table 10.1 we can calculate the expected lenght of our codes:

$$E[length\ of\ naive\ code] = \sum_{horse\ h=1}^{8} P(h\ horse\ winning) * length\ of\ code\ of\ h\ horse = 3$$

and

$$E[length\ of\ XX\ code] = \sum_{horse\ h=1}^{8} P(h\ horse\ winning) * length\ of\ code\ of\ h\ horse = 2$$

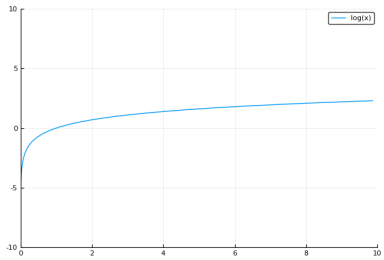
Later we will calculate the Entropy of x with the distribution from Table 10.1 and verify whether we have indeed found the most efficient encoding for x.

12.2 Explanation of H(x)

We can consider $H(x)$ in parts to try and gain some understanding. Again, with random variable x distributed according to $P(x)$:

$$H(x) = \sum_x P(x) log_2 \left(\frac{1}{P(x)} \right)$$

If we consider the first half of the equation, we can see that $\sum_x P(x)$ is the expected value of x. The second half of the expression for $H(x)$ includes the term $log_2 \left(\frac{1}{P(x)} \right)$



Notice that as $P(x)$ gets larger, $log_2 \left(\frac{1}{P(x)} \right)$ gets smaller.

The intution is that log_2 tells us the number of bits needed to express a number.
Here are some examples to help us find H(x) later on:

(a) Consider table 10.1 with x being horse 3.

$$\text{Then } \frac{1}{P(x)} = 8 \text{ and } \log_2 \left(\frac{1}{P(x)} \right) = \log_2(8) = 3$$

What that means is, we will need 3 bits of binary values to represent the number 8. No more, no less.

(b) Consider again table 10.1 with x being horse 5.

$$\text{Then } \frac{1}{P(x)} = 64 \text{ and } log_2(64) = 6.$$

Notice when the probability of x is larger, we will assign it fewer bits.

The length of a code can never be smaller than the entropy of the variable you are trying to encode. We can use this fact to verify whether the "XX Code" is in fact the most efficient possible code.

12.3 Finding H(x)

Example 12.1 (Horse Example)**.** Let x have the probability distribution described in Table 10.1. Then the Entropy of x is:

$$H(x) = \sum_x P(x) log_2 \left(\frac{1}{P(x)} \right) = \frac{1}{2} log_2 (2) + \frac{1}{4} log_2 (4) + \frac{1}{8} log_2 (8) + \frac{1}{16} log_2 (16) + \frac{1}{64} log_2 (64) + \frac{1}{64} log_2 (64) + \frac{1}{64} log_2 (64) = 2$$

Notice $E[length\ of\ XX\ code] = H(x) = 2$ which confirms that the "XX Code is the most efficient possible encoding for the data in Table 10.1.

Example 12.2 (Bernoulli($\frac{1}{4}$))**.** Recall if $f(x) = Bernoulli(p)$ then

$$f(x) = \begin{cases} 1 & \text{with probability p} \\ 0 & \text{with probability } 1-\text{p} \end{cases}$$

Let $x \sim Bernoulli(\frac{1}{4})$ then

$$H(x) = \frac{1}{4} * log_2(4) + \frac{3}{4} * log_2(\frac{4}{3}) = .811$$

Example 12.3 (Bernoulli($\frac{1}{2}$))**.** Let $x \sim Bernoulli(\frac{1}{2})$ then

$$H(x) = \frac{1}{2} * log_2(2) + \frac{1}{2} * log_2(2) = 1$$

Example 12.4 (Bernoulli($\frac{3}{4}$))**.** Let $x \sim Bernoulli(\frac{3}{4})$ then

$$H(x) = \frac{3}{4} * log_2(\frac{4}{3}) + \frac{1}{4} * log_2(4) = .811$$

Notice that Example 10.2 and Example 10.4 are identical. Let's consider one final example.

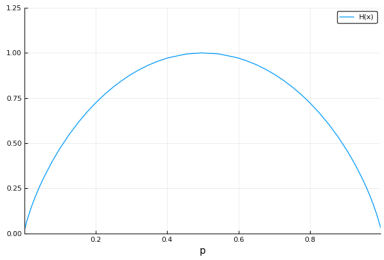
Example 12.5 (Bernoulli(1))**.** Let $x \sim Bernoulli(1)$ then

$$H(x) = P(1) * log_2\left(\frac{1}{1}\right) + P(0) * log_2\left(\frac{1}{0}\right)$$

Let's say that when $P(0)$ evaluates to 0 it "dominates" and we don't have to worry about the division by 0. So, $H(x) = 0$.

Now that we have seen so many Bernoulli examples, let's consider the plot of $H(x)$ and the parameter p from the Bernoulli distribution.

So, which is "more random", $Bernoulli(\frac{1}{4})$ or $Bernoulli(\frac{1}{2})$? $Bernoulli(\frac{1}{2})$ is in fact more random because each outcome is equally likely. With $Bernoulli(\frac{1}{4})$ we know that we should always guess $x = 0$ because $P(x = 0) = \frac{3}{4}$ We can also use Entropy to determine this. In the plot of $H(x)$, we see that for the Bernoulli distribution, $H(x)$ is greatest when $p = \frac{1}{2}$



12.4 How to Compute the Entropy of Data

Imagine we have a sequence of bits

$$1001010011100101000101001000001110010101$$

If we want to model this as a Bernoulli distribution, we don't know p but we can use an estimation.

$$number\ of\ ones\ = 17\ , number\ of\ zeros\ = 23\ , total = 40$$

So $\hat{\boldsymbol{P}}(x = 1) = \frac{17}{40}$ and $\hat{\boldsymbol{P}}(x = 0) = \frac{23}{40}$ and

$$\hat{\boldsymbol{H}}(x) = \frac{17}{40} * log_2\left(\frac{40}{17}\right) + \frac{23}{40} * log_2\left(\frac{40}{23}\right)$$

This is just an introductory example about calculating the Entropy of data. Further exploration is left to homework.