2. Suppose that female pandas who live beyond the age of 20 outnumber male pandas in the same age group by tree to one. How much information, in bits, is gained by learning that a panda that lives 20 is male?

$$h(\overset{\text{age}>20}{x}) = \lg\frac{1}{p(x)} = \lg\frac{1}{\frac{1}{4}} = \lg 4 = 2 \text{ bits} ✓$$

3. What is the maximum possible entropy H of an alphabet consisting of N different letters? In such a maximum entropy alphabet, what is the probability of its most likely letter? What is the probability of its least likely letter?

Maximum entropy of an alphabet Σ occurs when symbols are equiprobable. ✓ So the probability of any symbol (namely both the most and least likely symbols) is $\frac{1}{|\Sigma|} = \frac{1}{N}$. ✓

And the entropy is ...? (In terms of N.)

4. If discrete symbols from an alphabet S having entropy $H(S)$ are encoded into blocks of length n symbols, we derive a new alphabet of symbol blocks S^n. If the occurrence of symbols is independent, derive the entropy $H(S^n)$ of this new alphabet of symbol blocks.

The entropy is $H(S^n) = n \cdot H(S)$ ✓ I prove this result by induction. ✓

Case $n = 0$:

Trivial: we see that in any sequence of *no* symbols there is one result with $p = 1$ (namely the empty sequence) and therefore the entropy is $1 \cdot \lg 1 = 0 = 0 \cdot H(S)$. ✓

Oh, cool, yes.

Case $n = k + 1$:

With the induction hypothesis that $H(S^k) = k \cdot H(S)$.

$$\begin{aligned}
H(S^{k+1}) &= \sum_{\sigma^{k+1}\in S^{k+1}} p(\sigma^{k+1}) \cdot \lg\frac{1}{p(\sigma^{k+1})} && \text{Definition of entropy ✓}\\
&= \sum_{\sigma\in S}\sum_{\sigma^k\in S^k} p(\sigma\sigma^k) \cdot \lg\frac{1}{p(\sigma\sigma^k)} && \text{Definition of } S^{k+1} \text{ ✓}\\
&= \sum_{\sigma\in S}\sum_{\sigma^k\in S^k} p(\sigma)\cdot p(\sigma^k) \cdot \lg\frac{1}{p(\sigma)\cdot p(\sigma^k)} && \text{Independence ✓}\\
&= \sum_{\sigma\in S}\sum_{\sigma^k\in S^k} p(\sigma)\cdot p(\sigma^k) \cdot \lg\frac{1}{p(\sigma)} + p(\sigma)\cdot p(\sigma^k) \cdot \lg\frac{1}{\cdot p(\sigma^k)} \text{ ✓}\\
&= \left(\sum_{\sigma\in S} p(\sigma)\cdot\lg\frac{1}{p(\sigma)}\left(\sum_{\sigma^k\in S^k} p(\sigma^k)\right)\right) + \left(\sum_{\sigma\in S} p(\sigma)\right)\cdot\left(\sum_{\sigma^k\in S^k} p(\sigma^k)\cdot\lg\frac{1}{\cdot p(\sigma^k)}\right) \text{ ✓}\\
&= \left(\sum_{\sigma\in S} p(\sigma)\cdot\lg\frac{1}{p(\sigma)}\right) + \left(\sum_{\sigma^k\in S^k} p(\sigma^k)\cdot\lg\frac{1}{\cdot p(\sigma^k)}\right) && \text{Probabilities sum to 1 ✓}\\
&= H(S) + H(S^k) && \text{Definition of Entropy ✓}\\
&= (k+1)\cdot H(S) && \text{Induction Hypothesis ✓}
\end{aligned}$$

Thus, since we have that the theorem holds for $n = 0$ and if it holds at $n = k$ then it must also hold at $n = k + 1$; we can conclude that it must hold for *all* $n \in \mathbb{N}$. So we have proved the result, as required. Therefore $H(S^n) = n \cdot H(S)$ ✓

Excellent.

5. Why are fixed length codes inefficient for alphabets whose letters are not equiprobable? Discuss this in relation to Morse Code?

In this case, we have symbols which may contain very few shannon bits taking being represented with lots of bits. Consider the case of a random variable X ranging over an alphabet Σ with 128 symbols but $H(X) = 3$. If we use fixed-length codes, then we

Goal: reduce average codeword length.

have to use 7 bits to represent each symbol – where we could use a variable length code and have an average of $3 + \epsilon$ bits! ✓ Nice example.

Yeasss...

The entropy of letters in English (assuming independence) is ~ 4.1 bits per letter: if we use a fixed-length code then we require $\lceil \lg 26 \rceil = 5$ bits per letter. ✓

Morse Code is a code which exploits this difference: e is the most common letter of the alphabet and is represented with 1 dot: the shortest possible transmission. While z, the least common letter is represented with 5 dots. This make Morse Code more efficient than a fixed-length code. ✓ Yes, we really just have to define "efficiency"!

6. A fair coin is secretly flipped until the first head occurs. Let X denote the number of flips required. The flipper will truthfully answer any "yes-no" questions about his experiment, and we wish to discover thereby the value of X as efficiently as possible.

   (a) What is the most efficient possible sequence of such questions?

   Using the entropy-max principle, the most efficient questions to ask are those for which the answers are equiprobable – for every question.

   Therefore, the i^{th} question should be "was the first head at i" (starting at $i = 0$). This guarantees that the answer to the i^{th} question has a 50% chance of taking either answer. Therefore the information gained from each question is maximised. ✓

   (b) On average, how many questions should we need to ask?

   2 questions. ✓

   The entropy of X is 2 bits and each question gives 1 bit of information.

$$\forall x \in \mathbb{N}_{\geq 1}.\, P(x) = 2^{-x} \tag{1}$$

$$H(X) = \sum_{x=1}^{\infty} p(x) \cdot \lg \frac{1}{p(x)} \tag{2}$$

$$= \sum_{x=1}^{\infty} 2^{-x} \cdot \lg 2^{x} \tag{3}$$

$$= \sum_{x=1}^{\infty} x \cdot 2^{-x} \tag{4}$$

$$= \frac{1}{2} \cdot \sum_{x=1}^{\infty} x \cdot \frac{1}{2}^{x-1} \tag{5}$$

$$= \frac{1}{2} \cdot \frac{1}{\left(1 - \frac{1}{2}\right)^2} \tag{6}$$

$$= 2 \tag{7}$$

✓

   So the entropy of when the coin first landed heads is 2 bits and therefore, the number of questions we should need to ask on average is 2. ✓

   (c) Relate the sequence of questions to the bits in a uniquely decodable prefix code for X.

   The most efficient uniquely decodable prefix code for such a language is a code with variable sizes codes; where "heads on i^{th} throw is represented by $\underbrace{0 \ldots 0}_{\times i-1}1$.

   Each question we ask is checking to see whether the code representing that was the i^{th}. ✓ Good.

7. Is it possible to construct a prefix code in which the codewords have the following lengths: 1, 2, 3, 3, 4, 4.

Lets apply the Kraft inequality: for any uniquely decodable code over the alphabet Σ with codeword lengths $L = \{\ell_1, \dots \ell_n\}$

$$\sum_{\ell \in L} |\Sigma|^{-\ell} \leq 1$$

For the alphabet $\{0, 1\}$, apply this to the codewords we have:

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} = 1 + \frac{1}{8} > 1$$ ✓

So, we have that any code with codewords of those lengths over $\{0, 1\}$ does not satisfy the Kraft-inequality and therefore cannot be uniquely decodable. All prefix codes are uniquely decodable and therefore it cannot be a prefix code ✓

However, if we consider an alphabet $|\Sigma| > 2$ (it was never stated that we had to consider $\Sigma = \{0, 1\}$)then the inequality is satisfied and there is a uniquely decodable code (and a prefix code) with codewords of that length. ✓

Excellent.

8. Consider three variable-length codes for a four-symbol alphabet $\{A, B, C, D\}$ having probabilities $p(x)$ as shown:

x	$p(x)$	Code 1	Code 2	Code 3
A	0.25	00	0	01
B	0.5	1	0	0
C	0.125	01	110	011
D	0.125	10	111	111

Compare the average codeword length of each code to the entropy of the alphabet, and for each code give all possible decodings of the bit sequence "1001" as a complete message. Which codes are uniquely decodable; which have the prefix (instantaneous) property; which code is best, and why?

$$H(X) = \sum_{x \in X} p(x) \cdot \lg \frac{1}{p(x)} = 1.75 \text{ bits}$$ ✓

The fact that entropy has an operational definition through Shannon bits is critically important.

$$\bar{L_1} = 0.25 \cdot 2 + 0.5 \cdot 1 + 0.125 \cdot 2 + 0.125 \cdot 2 = 1.5 \text{ bits} \tag{8}$$
$$\bar{L_2} = 0.25 \cdot 2 + 0.5 \cdot 1 + 0.125 \cdot 3 + 0.125 \cdot 3 = 1.75 \text{ bits} \tag{9}$$
$$\bar{L_3} = 0.25 \cdot 2 + 0.5 \cdot 1 + 0.125 \cdot 3 + 0.125 \cdot 3 = 1.75 \text{ bits} \tag{10}$$ ✓

Code 1	BAB	DC
Code 2		
Code 3		

Table 1: Decodings of the bit sequence "1001"

✓

Code 2 and Code 3 are ✓ uniquely decodable and are compressed down to entropy. ✓ Code 2 is a prefix code. ✓ So Code 2 is the best code, since it is a prefix code at the entropy limit and none of the others are. ✓

↳ => instantaneously decodable also

9. Find a probability distribution $\{p_1, p_2, p_3, p_4\}$ such that there are two optimal codes that assign different lengths $\{\ell_i\}$ to the four symbols.

$\{\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}\}$. Consider the two codes $\{00, 01, 10, 11\}$ and $\{0, 10, 110, 111\}$. They both have a mean length of 2. ✓ Good.

10. Construct an ensemble where the difference between the entropy and the expected length of the Huffman code is as large as you can make it.

    Consider the ensemble (of length 1) $\{(a, 1-\delta), (b, \delta)\}$. Using a Huffman code, we get the code $\{(a, 0), (b, 1)\}$. ✓

    The difference between the entropy and the expected length of the Huffman code is given by:

    $$\Delta = ((1-\delta)\cdot 1 + \delta \cdot 1) - (1-\delta)\cdot \lg\frac{1}{1-\delta} - \delta \cdot \lg\frac{1}{\delta} \tag{11}$$
    $$= 1 - (1-\delta)\cdot \lg\frac{1}{1-\delta} - \delta \cdot \lg\frac{1}{\delta} \tag{12}$$
    $$\lim_{\delta \to 0} \Delta = 1 \tag{13}$$

    Yes, I'm not sure I agree that the limit of this quality is defined (since avg codeword length won't become 0).

    By the limits of Huffman Coding, we have that the expected length of a Huffman Code is *strictly less than* one greater than the entropy of the sequence. Thus, our code is the worst possible case. ✓

11. You are tasked with investigating a funky random number generator, which generates integers i where $1 \leq i \leq n$. The true distribution $(p_1, p_2, \ldots, p_n)$ is unknown, but the average μ is known. In order to perform inference and estimate the posterior distribution using Bayes theorem, we need a prior distribution.

    (a) Show, using the method of Lagrange multipliers, that for the maximum entropy prior p_i can be written as Cr^i for some C and r.

    Start by stating the function we wish to optimise and the constraints:

    $$H(X) = \sum p_i \cdot \lg\frac{1}{p_i}$$
    $$0 = \sum p_i - 1 \qquad \text{probabilities sum to 1}$$
    $$0 = \sum p_i \cdot i - \mu \qquad \mu \text{ is the mean}$$

    Now, we can form the Lagrangian

    $$L = H(X) + \lambda \cdot \left(\sum p_i - 1\right) + \eta \cdot \left(\sum p_i \cdot i - \mu\right)$$
    $$\frac{\partial L}{\partial p_i} = -\lg p_i - 1 + \lambda + \eta \cdot i$$

    ✓

    Now, recall that at maximum entropy, the derivative is zero

    $$0 = -\lg p_i - 1 + \lambda + \eta \cdot i$$
    $$p_i = 2^{\lambda - 1} \cdot (2^{\eta})^i \qquad \text{by rearranging}$$

    ✓

    Thus, we have that the maximum entropy prior p_i can be written as Cr^i for some C and r. ✓

    (b) Use normalisation to find C.

Using the fact that $\sum p_i = 1$ and these constants are shared between all p_i:

$$\sum_{i=1}^{n} p_i = 1 \qquad \text{probabilities sum to 1} \tag{14}$$

$$C \cdot \sum_{i=1}^{n} r^i = 1 \qquad \text{shown above} \tag{15}$$

$$C \cdot r \cdot \sum_{i=1}^{n} r^{i-1} = 1 \qquad \text{rearranging} \tag{16}$$

$$C \cdot r \cdot \frac{1-r^n}{1-r} = 1 \qquad \text{formula for sum of a geometric sequence} \tag{17}$$

$$C = \frac{1-r}{r(1-r^n))} \qquad \text{rearranging} \tag{18}$$

✓

Good.

Thus, we have that $C = \frac{1-r}{r(1-r^n)}$. ✓

(c) Use the known average of the distribution to obtain the following equation:

$$nr^{n+1} - (n+1)r^n + 1 = \mu(r^n - 1)(r - 1)$$

$$\mu = \sum_{i=1}^{n} p_i \cdot i \qquad \text{by definition of mean} \tag{19}$$

$$\mu = C \cdot \sum_{i=1}^{n} r^i \cdot i \qquad \text{expression for } p_i \tag{20}$$

$$\mu = \frac{1-r}{r(1-r^n)} \cdot \sum_{i=1}^{n} r^i \cdot i \qquad \text{expression for } C \tag{21}$$

$$\mu = \frac{1-r}{r(1-r^n)} \cdot r \cdot \sum_{i=1}^{n} r^{i-1} \cdot i \tag{22}$$

$$\mu = \frac{1-r}{r(1-r^n)} \cdot r \cdot \frac{\partial}{\partial r}\left(\sum_{i=1}^{n} r^i\right) \tag{23}$$

$$\mu = \frac{1-r}{1-r^n} \cdot \frac{\partial}{\partial r}\left(r \cdot \sum_{i=1}^{n} r^{i-1}\right) \tag{24}$$

$$\mu = \frac{1-r}{1-r^n} \cdot \frac{\partial}{\partial r}\left(\frac{r(1-r^n)}{1-r}\right) \qquad \text{sum of a geometric sequence} \tag{25}$$

$$\mu = \frac{1-r}{1-r^n} \cdot \frac{(1-(n+1)\cdot r^n)\cdot(1-r) + r\cdot(1-r^n)}{(1-r)^2} \tag{26}$$

$$\mu = \frac{1 - r - (n+1)\cdot r^n + (n+1)\cdot r^{n+1} + r - r^{n+1}}{(1-r^n)\cdot(1-r)} \tag{27}$$

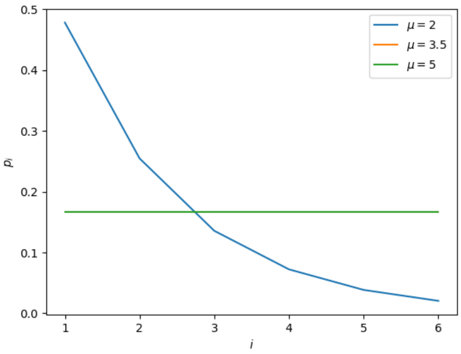
$$\mu = \frac{n \cdot r^{n+1} - (n+1)\cdot r^n + 1}{(1-r^n)\cdot(1-r)} \tag{28}$$

Finally, multiply both sides by $(1-r^n)\cdot(1-r)$ to get the required result:

$$n \cdot r^{n+1} - (n+1)\cdot r^n + 1 = \mu \cdot (1-r^n)\cdot(1-r)$$

✓

(d) This equation is difficult to solve analytically, so solve it numerically. For $n = 6$, plot the distribution of p_i for $\mu = 3.5$, $\mu = 2$ and $\mu = 5$.



✓ Nice. (Maybe better to use histogram or stem plot for discrete random variable?)

Figure 1: Prior Distribution according to Principle of Maximum Entropy

Notice that the distribution for $\mu = 3.5$ and $\mu = 5$ is the same! This is because, for $\mu \geq 3.5$ we have that $r \geq 1$. The equation is **not valid** for $r \geq 1$ because we had to use the sum of a geometric random variable in the derivation: and that equation is only valid for $r < 1$!

Hmm... wouldn't this just become $\frac{a(r^n - 1)}{r - 1}$ since n is finite, & rederive?