CS3236: Tutorial 2
(Symbol-Wise Source Coding)

Part I (Week 4) – Decodability, Kraft Inequality, and Shannon-Fano Coding

1. [**Unique Decodability**]

   (a) Is the code $\{00, 11, 0101, 111, 1010, 100100, 0110\}$ uniquely decodeable?

   (b) Is the ternary code $\{00, 012, 0110, 0112, 100, 201, 212, 22\}$ uniquely decodeable?

Symbol	Code $\mathcal{C}$	Code $\mathcal{D}$
a_1	1	10
a_2	10	1
a_3	100	01
a_4	1000	001

   (c) Is the code $\mathcal{C}$ shown above uniquely decodeable?

   (d) Is the code $\mathcal{D}$ shown above uniquely decodeable?

2. [**Shannon-Fano Code Example**]

   Let $\mathcal{X} = \{1, 2, 3, 4\}$, with the symbols having associated probabilities $\{0.4, 0.3, 0.2, 0.1\}$.

   (a) Compute the lengths of the Shannon-Fano code.

   (b) Construct a prefix-free code having those lengths.

   (c) Is this the optimal prefix-free code in terms of average length?

3. [**Optimal Prefix-Free Code**]

   Let the probability distribution of random variable X over an alphabet $\mathcal{X}$ be $P_X = \{p_1, p_2, ..., p_N\}$ where $N = |\mathcal{X}|$ and $p_1 < p_2 < .... < p_N$. Let $C(X)$ be the optimal prefix-free code with codeword lengths $\{\ell_1, \ell_2, ..., \ell_N\}$.

   (a) Show that the codeword lengths satisfy $\ell_1 \geq \ell_2 \geq .... \geq \ell_N$.
   (Hint: You might want to try a proof by contradiction.)

   (b) Further show that $\ell_1 = \ell_2$.
   (Hint: Again, proof by contradiction is useful.)

4. [**The Poisoned Glass**]

   'Mathematicians are curious birds', the police commissioner said to his wife. 'You see, we had all those partly filled glasses lined up in rows on a table in the hotel kitchen. Only one contained poison, and we wanted to know which one before searching that glass for fingerprints. Our lab could test the liquid

in each glass, but the tests take time and money, so we wanted to make as few of them as possible by simultaneously testing mixtures of small samples from groups of glasses. The university sent over a mathematics professor to help us. He counted the glasses, and thought of doing a binary search strategy,[1] smiled and said: "Pick any glass you want, Commissioner. We'll test it first."
"But won't that waste a test?" I asked.
"No," he said, "There are 65 glasses, and 64 is a much nicer number to work with. Let's test a one glass first – it doesn't matter which one – and if it doesn't turn out to be the poisoned one, then we can test the rest using a binary search." '

Create a strategy suggested by the mathematics professor which includes picking the first glass and testing it on its own, followed by a binary search on the remaining 64 if needed. Also, consider an alternative strategy that does binary search right from the start with rounding, splitting 65 glasses into groups of size 32 and 33, then either 33 into 16 and 17 **or** 32 into 16 and 16, and so on.

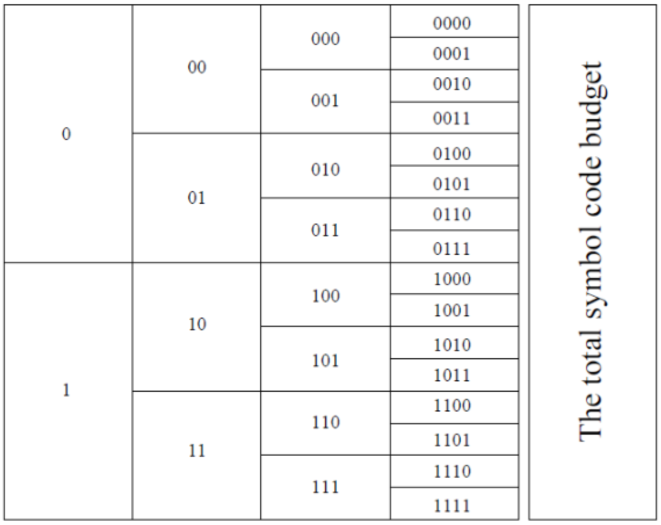
Calculate the average number of tests of each strategy, assuming the poison was placed uniformly at random into one of the 65 glasses. What does this have to do with source coding?

5. [**Proof of Existence Theorem for Kraft's Inequality**]

   In the Lecture, we have discussed that for any uniquely decodable code $C(X)$ for a random variable X over the alphabet $\mathcal{X}$, the codeword lengths must satisfy: $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$.

   For notational convenience, let $\mathcal{X} = \{1, \ldots, N\}$, and denote the corresponding lengths by $\{\ell_1, \ldots, \ell_N\}$. Prove that if these lengths satisfy $\sum_{i=1}^{N} 2^{-\ell_i} \leq 1$, then there exists a prefix free code with these codeword lengths $\{\ell_1, \ldots, \ell_N\}$.

   (Hint: Try to use Figure 1 as a guide and show that suitable codeword allocations can be done without running out of space. Start with the shortest codewords. You could try a few simple examples before trying the general case.



The codeword supermarket and the symbol coding budget. The 'cost' 2^{-l} of each codeword (with length l) is indicated by the size of the box it is written in. The total budget available when making a uniquely decodeable code is 1.

Figure 1: Illustration of the "supermarket budget".

[1] Binary search works as follows: Split the N glasses into 2 groups of size $N/2$ (or sizes $\lfloor N/2 \rfloor$ and $\lceil N/2 \rceil$ if N is an odd number). Mix all the liquids in one group together and test it – if it's poisonous, we know that group has the poison glass, and otherwise, it's the other group that has it. Recursively apply this procedure on the group with the poison until only one glass remains. It is easy to show that this requires $\lceil \log_2 N \rceil$ tests.

6. [**Shannon-Fano Code Length Bound**] It is known that the Shanon-Fano code achieves an average length $L(C)$ satisfying
$$H(X) \leq L(C) < H(X) + 1,$$
for *any* source distribution P_X. Explain why when making such a claim for arbitrary P_X, we cannot get a better (i.e., smaller) constant than 1 on the right-hand side, *even if we use an optimal symbol-wise code.*

Hints for Part I

1. Things to look for: Counter-examples, prefix-free, Kraft inequality
2. Fairly basic example – check the lecture notes if any notions are unclear.
3. In (a), if $\ell_i < \ell_j$, what happens if we swap the associated codewords? In (b), if $\ell_1 > \ell_2$, consider a trick for making the code shorter while remaining prefix-free.
4. The professor's strategy is fairly easy to analyze because $64 = 65 - 1$ is a power of two. For the other strategy, show that most of the time we need 6 tests, but there are two cases (out of 65) in which we need 7 tests.
5. Given the lengths, try assigning the shortest codewords (most probable inputs) first and things should work out relatively neat. You could check on a few small examples before arguing about the general case.
6. Think about low-entropy sources.

Part II (Week 5) – Huffman Coding

7. [**Optimal Codes with Different Lengths**]

   Find a probability distribution $\{p_1, p_2, p_3, p_4\}$ such that there are two optimal prefix-free codes whose *unordered lists of lengths* assigned are different (e.g., lengths $1, 2, 3, 4, 4$ and $4, 1, 4, 2, 3$ would be considered the same "list of lengths" because they are the same numbers in a different order, whereas lengths $1, 2, 3, 4, 4$ and $2, 2, 2, 3, 3$ would be considered two different lists of lengths).

   (Hint: Try the Huffman algorithm and see if two solutions arise.)

8. [**(Im)possible Huffman Codes**]: Which of these codes cannot be Huffman codes for any probability assignment?

   (a) $\{0, 10, 11\}$.

   (b) $\{00, 01, 10, 110\}$.

   (c) $\{01, 10\}$.

9. [**Huffman Code Length**]

   Consider a source X with four symbols $\mathcal{X} = \{a_1, a_2, a_3, a_4\}$ and probability distribution $P_X = \{p_1, p_2, p_3, p_4\}$. Let ℓ_i be the length of the codeword associated with symbol a_i, generated by a Huffman code applied to the source X.

   Suppose $p_1 \geq p_2 = p_3 = p_4$. Find the smallest value of q such that $p_1 > q$ implies $\ell_1 = 1$.

10. [**Huffman Code with Ideal Code Lengths**]

    Consider a source X with n symbols $\mathcal{X} = \{a_1, a_2, a_3..., a_n\}$ and probability distribution $P_X(a_i) = 2^{-i}$ for $0 < i < n$ and $P_X(a_n) = 2^{-n+1}$. Let l_i be the length of the codeword associated with symbol a_i, generated by a Huffman code applied to the source X.

(a) Show that the entropy of the source X is given by $2(1 - 2^{-n+1})$.
(Hint: The identity $\sum_{i=1}^{i=n-1} 2^{-i} i = 2^{-n+1} \cdot (2^n - n - 1)$ is useful.)

(b) Give one explicit form for the codeword associated with symbol a_i, generated by a Huffman code. In other words, what is the length of the codeword associated with symbol a_i?

(c) Calculate the expected length of the Huffman code for the source X that you mention in part (b). Compare with the entropy of the source X.

11. [**Yes/No Question Game**]

Let the random variables D_1 and D_2 be independent and identically distributed uniformly in $\{1, 2, 3, 4, 5, 6\}$. Let $D = |D_1 - D_2|$. Alice knows D. Your task is to ask Alice a sequence of questions with yes/no answers to find out D. For instance, you may ask questions like "Is D equal to 0?" and "Is D one of the values in $\{0, -2, 3\}$?".

Devise a strategy that achieves the minimum possible average number of questions. Calculate the average number of questions for your strategy.

(Hint: This is source coding in disguise.)

12. [**Huffman Code Lengths**]

Let $p_1 > p_2 > p_3 > p_4$ be the symbol probabilities for a source alphabet size $N = |\mathcal{X}| = 4$, and let the correspond codeword lengths be $\ell_1, \ell_2, \ell_3, \ell_4$.

(a) What are the possible sets of codeword lengths $\{\ell_1, \ell_2, \ell_3, \ell_4\}$ for a Huffman code for the given type of source?

(b) Suppose that $p_1 > p_3 + p_4$. What are the possible sets of codeword lengths now?

(c) What are the possible sets of codeword lengths if $p_1 < p_3 + p_4$?

(d) **(Harder)** What is the smallest value of ρ such that $p_1 > \rho$ implies that $\ell_1 = 1$? *(Hint: What are the largest values of p_1, p_3, p_4 for which part (b) holds?)*

(e) **(Harder)** What is the largest value of λ such that $p_1 < \lambda$ implies $\ell_1 = 2$?

Hints for Part II

7. This only happens when some Huffman step "connect the two smallest probabilities" has a tie.

8. If "yes" then construct an example of a distribution generating that code. The "no" case(s) can be proved by showing that the code is suboptimal, whereas Huffman code is known to be optimal.

9. Step through the Huffman algorithm in generic notation, then see how various q values would behave at the relevant step.

10. (a) is a direct calculation along with the hint. In (b) try a few small examples as a hint towards the general case. In (c) just compute the average based on your answer to part b.

11. Find the PMF on D and argue why its Huffman code is relevant here.

12. In (a) try to rule out certain cases (e.g., that the smallest length is ≥ 3) using the fact that the Huffman code is optimal. In (b) and (c) check what happens specifically in the second step of the Huffman algorithm. In (d) assume $p_1 > \rho$, then argue that $p_3 + p_4 < f(\rho)$ for some $f(\cdot)$, then solve the equation $\rho = f(\rho)$ to get $\rho = 2/5$. In (e) assume $p_1 < \lambda$, then show that $p_3 + p_4 > g(\lambda)$, and then solve the equation $\lambda = g(\lambda)$ to get $\lambda = 1/3$.