The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE300 Communication Theory
Problem Set V: Information Theory
October 4, 2021

Unless specified otherwise, compute entropy in bits. You may use MATLAB or a calculator to help with the number crunching. You don't need to show MATLAB code if you just used it for simple numeric calculations (like computing $-\sum p \log p$). But if it is easier to give me (clear) code rather than writing out by hand that is OK, as long as you actually give me the answers (or code that displays the answers properly labeled when I run it). MAKE SURE ANY CODE YOU GIVE ME IS EXECUTABLE.

1. Consider a DMS with symbols from the alphabet $\{abcdefgh\}$ with respective probabilities:
$$\{0.24, 0.21, 0.18, 0.15, 0.07, 0.06, 0.05, 0.04\}$$
   (a) Consider a VLC where the code lengths are related to the probabilities via:
   $$\ell_i = \lceil -\log_2 p_i \rceil$$
   where $\lceil x \rceil$ denotes the smallest integer $\geq x$. Show that the lengths chosen this way satisfy the Kraft inequality.
   (b) Compute the entropy of the source $H(X)$, and the average code length $\bar{L}$, using the proposed lengths. Verify that $H(X) \leq \bar{L} < H(X) + 1$.

2. For the symbol probabilities given in the previous problem, construct a Huffman code. Show the code tree and give the code table. Find the code lengths, and compare them to the code lengths proposed in the previous problem. If they differ, repeat the computations: confirm the Kraft inequality, compute $\bar{L}$ and compare with $H(X)$.

3. For the Huffman code you performed above, let $Y_1 Y_2 Y_3 \cdots$ denote the sequence of coded binits.
   (a) Compute $H(Y_1)$, $H(Y_2)$ and $H(Y_2|Y_1)$. **Note:** You should report values with 4 decimal place precision!
   (b) Based on your results to part (a), we can definitively say $Y_1 Y_2 Y_3 \cdots$ is **not** a DMS. What SPECIFICALLY leads to that conclusion?
   (c) Based on your results to part (a), we can definitively say $Y_1 Y_2 Y_3 \cdots$ is **not** a stationary source. What SPECIFICALLY leads to that conclusion?

4. Recall the rate-distortion function for a Gaussian source with variance σ^2 (using square error as the distortion) is:

$$R_g(D) = \begin{cases} \frac{1}{2}\log_2(\sigma^2/D) & D \le \sigma^2 \\ 0 & D > \sigma^2 \end{cases}$$

You will explore the water filling algorithm in this problem. Let $\vec{X}$ be a real Gaussian vector with N independent components with respective variances σ_i^2. As you recall in the water filling algorithm, we select a fixed λ, and assign distortion to the i^{th} component $D_i = \lambda$ if $\sigma_i^2 > \lambda$, and otherwise $D_i = \sigma_i^2$. The corresponding rate will be $R_i = R_g(\lambda)$ or $R_i = 0$, respectively. With total distortion $D = \sum D_i$, $R = \sum R_i$, the resulting (R, D) will be optimal. The problem is we do not know in advance which value of λ will yield a desired value for D or for R. We basically have to iterate through several choices of λ to try to achieve a desired target value for D or R.

Here, consider the following situation:

- We have $\sigma_1^2 = 1$, $\sigma_2^2 = 0.8$, $\sigma_3^2 = 0.1$, $\sigma_4^2 = 0.01$.
- We want $R \le 3$.

The idea is to iterate over λ so we get R close to, but not over, 3. The first thing to recognize is that decreasing λ increases R (for example, if $\lambda > \max \sigma_i^2$ then we get $R = 0$!). We want to achieve a condition where λ_1 gives $R < 3$ and λ_2 gives $R > 3$. Then try the midpoint value $(\lambda_1 + \lambda_2)/2$, and iterate. That is, at each step, we have two choices for λ that straddle the target value for R. This is called the bisection method.

Either do this "manually" with MATLAB, iterating until you get $2.9 < R \le 3$, that is you are using MATLAB as a calculator; or, if you are ambitious, write code to implement this algorithm, and run it until you get $2.9 < R \le 3$. If this happens too quickly, then go through at least 3 iterations and see how close you get. Once you find your final choice for λ, specify the (R, D) point. **Remark:** I do not want you to attempt to use a "canned" nonlinear equation solver to find λ and (R, D) directly. I want you to use the iterative procedure described here.

5. Find the transmission power (in Watts) necessary for an AWGN channel with $N_0 = 10^{-8} Joules$, transmission bandwidth $B = 1MHz$, to achieve a channel capacity of $1Mbps$.