EE276: Homework #3

Due on Friday Jan 31, 6pm - Gradescope entry code: 5K35EZ

1. **Bad codes.**
   Which of these codes cannot be a Huffman code for any probability assignment? Justify your answers.

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

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

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

2. **Coin Toss Experiment and Golomb Codes**

   Kedar, Mikel and Naroa have been instructed to record the outcomes of a coin toss experiment. Consider the coin toss experiment $X_1, X_2, X_3, ...$ where X_i are i.i.d. $Bern(p)$ (probability of a H (head) is p), $p = 15/16$.

   (a) Kedar decides to use Huffman coding to represent the outcome of each coin toss separately. What is the resulting scheme? What compression rate does it achieve?

   (b) Mikel suggests he can do a better job by applying Huffman coding on blocks of r tosses. Construct the Huffman code for $r = 2$.

   (c) Will Mikel's scheme approach the optimum expected number of bits per description of source symbol (coin toss outcome) with increasing r? How does the space required to represent the codebook increase as we increase r?

   (d) Naroa suggests that, as the occurrence of T is so rare, we should just record the number of tosses it takes for each T to occur.
   To be precise, if Y_k represents the number of trials until the k^{th} T occurred (inclusive), then Naroa records the sequence:

   $$Z_k = Y_k - Y_{k-1}, \; k \geq 1, \tag{1}$$

   where $Y_0 = 0$

   i. What is the distribution of Z_k, i.e., $P(Z_k = j), j = 1, 2, 3, ...$?

   ii. Compute the entropy and expectation of Z_k.

   iii. How does the ratio between the entropy and the expectation of Z_k compare to the entropy of X_k? Give an intuitive interpretation.

   (e) Consider the following scheme for encoding Z_k, which is a specific case of Golomb Coding. We are showing the first 10 codewords.

Z	Quotient	Remainder	Code
1	0	1	1 01
2	0	2	1 10
3	0	3	1 11
4	1	0	0 1 00
5	1	1	0 1 01
6	1	2	0 1 10
7	1	3	0 1 11
8	2	0	00 1 00
9	2	1	00 1 01
10	2	2	00 1 10

i. Can you guess the general coding scheme? Compute the expected codelength of this code.

ii. What is the decoding rule for this Golomb code?

iii. What makes this codebook suitable for efficient encoding and decoding?

3. **Arithmetic Coding.**

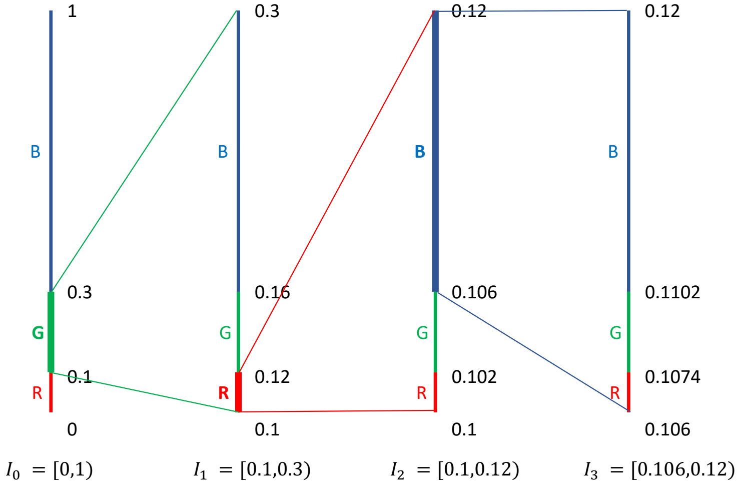

Figure 1: Illustration of arithmetic coding.

Note: Throughout this problem, we will work with digits rather than bits for simplicity. So the logarithms will be base 10 and the compressor will output digits $\{0, 1, \ldots, 9\}$.

This problem introduces a simplified version of arithmetic coding, which is itself based on Shannon-Fano-Elias coding. Arithmetic coding takes as input a sequence $x^n \in \mathcal{X}^n$ and a distribution q over $\mathcal{X}$. The encoder maintains an interval which is transformed at each step as follows:

- Start with $I_0 = [0, 1)$.
- For $i = 1, \ldots, n$:

- Divide I_{i-1} into $|\mathcal{X}|$ half-open subintervals $\{I_{i-1}^{(x)},\ x \in \mathcal{X}\}$ with length of $I_{i-1}^{(x)}$ proportional to $q(x)$, i.e., $\left|I_{i-1}^{(x)}\right| = q(x)\,|I_{i-1}|$ for $x \in \mathcal{X}$.
- Set $I_i = I_{i-1}^{(x_i)}$

Figure 1 shows an example of this for $\mathcal{X} = \{R, G, B\}$, $(q(R), q(G), q(B)) = (0.1, 0.2, 0.7)$ and $x^3 = GRB$. At the end of this process, the encoder selects a number in the interval I_n and outputs the digits after the decimal point for that number. In the example shown, the encoder can output 11, which corresponds to $0.11 \in [0.106, 0.12)$. While 1103 (corresponding to 0.1103) is also a valid output, the encoder tries to output the shortest possible valid sequence. The YouTube video `https://youtu.be/FdMoL3PzmSA` might be helpful for understanding this process even better.

(a) Briefly explain how the decoding might work in a sequential manner. You can assume that the decoder knows the alphabet, the distribution q and the length of the source sequence n.

(b) What is the length of interval I_n in terms of q and x^n?

(c) For the following intervals I_n obtained by following the above-described process for some x^n, find the length of the shortest output sequence (in digits):

i. $[0.095, 0.105)$

ii. $[0.11, 0.12)$

iii. $[0.1011, 0.102)$

In general, if the interval length is l_n, then the shortest output sequence has at most $\left\lceil \log \frac{1}{l_n} \right\rceil$ digits.

(d) Show that the length $l(x^n)$ for the arithmetic encoding output satisfies

$$l(x^n) \leq \log \frac{1}{q(x_1)\dots q(x_n)} + 1$$

(e) Suppose that $X^n \overset{\text{i.i.d.}}{\sim} X$ which has PMF P, and we use arithmetic coding with $q = P$. Then show that

$$\frac{1}{n} E[l(X^n)] \leq H(X) + \frac{1}{n}$$

Compare this to Huffman coding over blocks of length n with respect to compression rate and computational complexity.

(f) Suppose both the encoder and the decoder have a prediction algorithm (say a neural network) that provides probabilities $q_i(x|x^{i-1})$ for all i's and all $x \in \mathcal{X}$. How would you modify the scheme such that you achieve

$$l(x^n) \leq \log \frac{1}{q_1(x_1) q_2(x_2|x_1) \dots q_n(x_n|x^{n-1})} + 1$$

Thus, if you have a prediction model for your data, you can apply arithmetic coding on it - good prediction translating to high probability, in turn translating to short compressed representations.

4. **Entropy Rate.**
Consider the Markov process from class taking values in $\{H, T\}$ with the joint probability distribution given as

$$P(X_1 = x_1, \ldots, X_n = x_n) = P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})$$

where $P(X_1 = H) = \frac{1}{2}$, $P(X_i = H|X_{i-1} = H) = \frac{3}{4}$ and $P(X_i = T|X_{i-1} = T) = \frac{3}{4}$ for all $i > 1$.

   (a) Directly compute $P(X_2 = H)$ and extend that result to show that the process is stationary (we are only looking for the main idea, no need to write a long proof).

   (b) Compute $H(X_n|X_{n-1}, \ldots, X_1)$ as a function of n and find the limit as $n \to \infty$.

   (c) Compute $\frac{1}{n}H(X_1, \ldots, X_n)$ as a function of n and find the limit as $n \to \infty$. How does this relate to the result in part (b)?

5. **Individual Sequences and a Universal Compressor.**
Note: Ignore integer constraints on codeword lengths throughout this problem.
Notation: $h_2(p) = -p \log_2 p - (1-p) \log_2(1-p)$ (= binary entropy function).

Let x^n be a given arbitrary binary sequence, with n_0 0's and n_1 1's ($n_1 = n - n_0$). You are also provided a compressor C_q which takes in any arbitrary distribution q on $\{0, 1\}$ as a parameter, and encodes x^n using:

$$\bar{L}_q(x^n) = \frac{1}{n} \log \frac{1}{q(x^n)}$$

bits per symbol where $q(x^n) := \prod_{i=1}^{n} q(x_i)$.

   (a) Given the sequence x^n, what binary distribution $q(x)$ will you choose as a parameter to the compressor C_q, so that $\bar{L}_q(x^n)$ is minimized. Your answer (values of $q(0)$ and $q(1)$) will be expressible in terms of n, n_0 and n_1.

   (b) When compressing any given individual sequence x^n, we also need to store the parameter distribution $q(x)$ (required for decoding). Show that you can represent the optimal parameter distribution $q(x)$ from part (a) using $\log(n + 1)$ bits. You can assume that the decoder knows the length of the source sequence n.

   (c) Show that the effective compression rate for compressing x^n (in bits per source symbol) with the distribution q from part (a) is $h_2(n_1/n) + \log(n + 1)/n$.

   (d) Now suppose that we apply the scheme above to X^n sampled from an i.i.d. $Ber(p)$ distribution. Show that the expected compression rate approaches $h_2(p)$ as $n \to \infty$, i.e., the scheme is a *universal* compressor for i.i.d. sources.

6. **Extending to Shannon Codes**
For a general source, let

$$n_u^* = \lceil \log \frac{1}{p(u)} \rceil \qquad \forall u \in \mathcal{U}.$$

Then,

$$\sum_{u \in \mathcal{U}} 2^{-n_u^*} \leq 1.$$

We want to consider a new source $p^*(u) = 2^{-n_u^*}$. $p^*(u)$ does not sum to 1 over $\mathcal{U}$, but we claim that we can add a **finite** number of new symbols to extend the source to $\mathcal{U}^* \supseteq \mathcal{U}$ such that $p^*(u)$ is dyadic over $\mathcal{U}^*$. Prove this claim.

Hint: How can you reduce this problem to showing that certain rational numbers have a finite binary representation?

7. **Decoding LZ77**

We encoded a binary sequence using LZ77; we now want to decode the resulting bitstream. We first decode it into the triplets and obtain:

$$\begin{array}{cccc} (0,\ 0,\ 1) & (0,\ 0,\ 0) & (1,\ 5,\ 1) & (8,\ 2,\ 1) \\ \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)} \end{array}$$

Recall that the first entry of the triplet indicates how far back in the sequence you must go to start decoding the phrase; the second entry of the triplet indicates how many elements from that point should be "copied" into your newest phrase entry; and the final entry of the tuple indicates the new element (unseen in the past sequence) that should be added.

Specify how these triplets will now be decoded to reconstruct the original source sequence.