Problem Set #5
Due: Friday, April 17, 2015

1. **Huffman codes and entropy:**

   (a) Construct a Huffman code for the probability distribution

   $$p = (.3,\ .14,\ .13,\ .12,\ .1,\ .06,\ .05,\ .05,\ .01,\ .01,\ .01,\ .01,\ .01).$$

   What is the average length of this code? What is the entropy of the distribution p?

   (b) Find a probability distribution that, when used with the code constructed above, has average length equal to its entropy.

2. **Bad Codes:** (from Cover-Thomas) Which of these codes cannot be Huffman codes for any probability assignment and why?

   (a) $\{0, 10, 11\}$
   (b) $\{00, 01, 10, 110\}$
   (c) $\{01, 10\}$

3. **Shannon codes and Huffman codes:** (from Cover-Thomas) Consider a random variable X that takes on four values with probabilities (1/3, 1/3, 1/4, 1/12).

   (a) Construct a Huffman code for this random variable.
   (b) Show that there exist two different sets of optimal lengths for the codewords; namely, show that codeword length assignments $(1, 2, 3, 3)$ and $(2, 2, 2, 2)$ are both optimal.
   (c) Conclude that there are optimal codes with codeword lengths for some symbols that exceed the Shannon code length. (What are the Shannon code lengths for this distribution?)

4. Estimate the number of bits of information (entropy) in each of the following observations. Briefly justify your answers.

   (a) The day of the maximum temperature in Princeton in the coming July.
   (b) Whether or not it snows in Princeton on the day in part (a).
   (c) A random person's birthday.
   (d) Among 10 twenty-year-old students, whether or not any of them were born on the same day.
   (e) Among 23 twenty-year-old students, whether or not any of them were born on the same day. (search online for "birthday problem" for a hint)
   (f) Among 40 twenty-year-old students, whether or not any of them were born on the same day.

5. **Entropy of English:**

   (a) Find a table online of letter frequencies for the English language (there is a Wikipedia page called "letter frequency") and calculate the entropy. You can ignore space and punctuation, even though this does have a small effect on the entropy.

   (b) If the 26 letters appear with equal probability, recalculate the entropy.

   (c) The true *entropy rate* of English, which gives the fundamental lower bound for compression (allowing for compression of blocks of letters), is estimated to be about 1 bit per letter. Can you explain the difference between this and your answer to part (a)?

   (d) Find an English text from the web (no less than 1000 words—specify your source of text) and count the total number of letters, n. Then compress the .txt. file into .zip or .rar (specify your compression software), and check the number of bits, m, for this file. What is the ratio m/n? Is this compression software performing optimally? (note: These compression algorithms do not benefit from knowing that the file is English text. They are universal and simply look for patterns in the data to exploit for compression. Their performance improves as the size of the data increases.)

6. **Hamming codes for error correction:**

   (a) Hamming codes correct single bit errors. For any integer k there is a Hamming code of size $n = 2^k - 1$ which consists of k bits of redundancy. What is the rate of Hamming codes as a function of n (rate means bits of message per bit of transmission)? What constant does this approach as n goes to infinity?

   (b) Suppose each transmitted bit has probability p of being received in error. What is the probability that the Hamming code is decoded in error, as a function of n? (Hint: The probability of error is one minus the probability of being correct. For correct decoding there must be either no bit errors or one error, which can occur in n possible places. Add the contributions of the probability of each of these sequences. For example, for the no error sequence, the probability is $(1-p)^n$) What constant does this approach as n goes to infinity?

7. **One-time pad:** Apply a one-time pad to the message $m = 0110100101$ using the key $k = 0011011010$.

8. **SNR from quantization:**

   (a) Suppose a signal with values $x[n] \in [-1, 1]$ is uniformly quantized with b bits. Calculate the signal-to-noise ratio (SNR) of the quantized signal (power of the signal divided by power of the noise). For simplicity, assume that the samples $x[n]$ are uniformly distributed on the range $[-1, 1]$.

   (b) It's common to measure ratios of powers on a log-scale using decibels, especially for signal-to-noise ratios. A decibel ($1dB$) is $10 \log_{10}$ of the ratio. Calculate the SNR of the quantized signal in dB. What is the value of each quantization bit in terms of dB?

9. **Circular Convolution Property:** Consider the following two finite duration discrete-time signals.

$$x[0] = 1, \quad x[1] = 1, \quad x[2] = 0, \quad x[3] = 0.$$
$$y[0] = 0, \quad y[1] = 1, \quad y[2] = 0, \quad y[3] = 1.$$

   (a) Calculate the circular convolution of $x[n]$ and $y[n]$.

   (b) Calculate the DFT of $x[n]$ and $y[n]$, multiply, and calculate the inverse DFT.

   Note: Make sure to use the DFT rather than the DTFS (the difference is the normalization constant.). If you use the DTFS then you need to include the factor of $N = 4$ is the convolution property.

10. Problem 5.28 from the textbook.

11. Problem 7.27 from the textbook.

    Remember that the textbook substitutes $\omega = 2\pi f$. The problems might make more sense if you make the substitutions back to f.

    Also, the textbook uses the notation $X(j\omega)$ to mean the continuous-time Fourier transform of $x(t)$ and $X(e^{j\omega})$ for the discrete-time Fourier transform of $x[n]$, where we have been using $X(f)$ in both cases.