ECE 563 Spring 2016 **Homework 3**

Date Assigned: 23 September 2016.
Date Due: 4 October 2016 in class.

1. Under what conditions on g and the joint distribution of (X, Y) is $H(X|Y)$ exactly equal to $H(X|g(Y))$?

2. Suppose X takes on the first m counting numbers with decreasingly ordered probabilities $p_1 \geq p_2 \geq \cdots \geq p_m$. The minimal probability of error predictor of X is $\hat{X} = 1$ since this is the most likely element. The corresponding probability of error $p_e = 1?p_1$. Consider the space of all probability mass functions $p = (p_1, \ldots, p_m)$ (again, decreasingly ordered). Maximize $H(p)$ over this space subject to the constraint $1 - p_1$ (the error probability) is fixed (to say, p_e). Your answer should be in terms of p_e. Reflect on how this is connected to Fano's inequality.

3. Which of the following inequalities are generally $\geq, =, \leq$? Label each of the below with $\geq, =, \leq$.

   (a) $H(5X)$ vs $H(X)$.

   (b) $H(X2|X1)$ vs $H(X2|X1, X3))$.

   (c) $H(X, Y)$ vs $H(X) + H(Y)$.

   (d) $I(g(X); Y)$ vs $I(X; Y)$.

4. We wish to use a 3-sided coin X to generate a fair coin toss. The probability mass function of X is $P(X = a) = p_a$, and $P(X = b) = p_b$, and, $P(X = c) = p_c$, where p_a, p_b, p_c are unknown (except that they sum to 1).

   (a) How would you use two independent flips X_1, X_2 to generate (if possible) a Bernoulli(0.5) random variable?

   (b) What is the resulting maximum expected number of fair bits generated?

   We will revisit this problem later in the course on extracting pure randomness from unknown sources.

5. In this question, you are asked to implement on Matlab the iterative decoding of Fountain codes, a class of codes for the binary erasure channel. You should answer each part of the question, and hand in any Matlab code and plots.

(a) A binary linear code consists of K information bits and N coded symbols, each of which is a modulo 2 sum of a subset of the information bits. Design a data structure for representing the code. The data structure should be both space-efficient and easy to update by the iterative decoding algorithm. You can assume that the number of information bits that each coded symbol depends on is bounded by D, and D is much smaller than N and K (low-density).

(b) A codeword that is encoded via a linear code can be decoded at the output of an erasure channel simply by inverting the submatrix of the linear code matrix that corresponds to the non-erased codeword symbols. But inverting a matrix is computationally expensive ($O(K^3)$) and it is natural to look for simpler decoding procedures. Iterative decoding is one such procedure and has only linear time complexity. A short summary is given below:

i. Step 1: Look for those output symbols that are not erased and are degree one coded. This makes for easy decoding of the single information bit involved.

ii. Step 2: Now re-represent the code based on the fact that some information bits have already been decoded. (This is where the smart data structure you designed in the earlier part comes in handy.) Go back to Step 1.

Test your algorithm on the following simple example: three information bits $b_1; b_2; b_3$. Six coded symbols: $x_1 = b_1; x_2 = b_2; x_3 = b_3; x_4 = b_1 + b_2 + b_3; x_5 = b_2 + b_3; x_6 = b_1 + b_2$. Only the second and third coded symbols are erased.

(c) Now consider the fountain code, where each coded symbol is independently and randomly generated. Use the following weight distribution to generate the code at each time instant: Ω_i is the probability that i information bits (randomly chosen) are combined to form the coded symbol: $\Omega_1 = 0.007969; \Omega_2 = 0.493570; \Omega_3 = 0.166220; \Omega_4 = 0.072646; \Omega_5 = 0.082558; \Omega_8 = 0.056058; \Omega_9 = 0.037229; \Omega_{19} = 0.055590; \Omega_{65} = 0.025023; \Omega_{66} = 0.003135$. Suppose $K = 5000$. And suppose the destination receives 100 such coded symbols (so $N = 100$). Run the iterative decoding algorithm. How many information bits can you decode before the algorithm stalls? Give a simple upper bound on the maximum possible number of bits that can be decoded from the 100 symbols.

(d) Now keep the 100 symbols already received and suppose you receive an additional 100 coded symbols. Run the iterative decoding algorithm on the 200 coded symbols. How many information bits can be decoded, and compare to the upper bound.

(e) Repeat previous part, adding 100 more symbols each time, until all the 5000 information are decoded. Plot the number of information bits decoded as a function of the number of received coded symbols, always comparing it to the upper bound. Comment on the performance of the algorithm, both in terms of the progress it makes as it receives more coded symbols, and in terms of the number of coded symbols needed to decode all the bits.

6. Let $\{X_n\}_{n=0}^{\infty}$ be a stationary binary Markov chain with transition probability matrix as follows: $P(X_k = 0|X_{k?1} = 0) = P(X_k = 1|X_{k?1} = 1) = 0.75$. Calculate the first 3 bits of $F(X^{\infty}) = 0.F_1F_2\cdots$ when $X^{\infty} = 0.101011111\ldots$. How many bits of X^{∞} does this specify?

7. These questions relate to the LZ78 compression algorithm presented in a paper by Lempel and Ziv in 1978. Each question is to be answered independent of the others. A nice reference for LZ78 is this lecture notes by Prof. Peter Shor at MIT:
`http://www-math.mit.edu/~djk/18.310/Lecture-Notes/LZ-worst-case.pdf`

   (a) Give the LZ78 parsing and encoding of 000000110101000000110101.

   (b) We are given the constant sequence $x^n = 1111111\cdots1$ (i.e., we have n consecutive 1's).

      i. Give the LZ78 parsing of this sequence.

      ii. Argue that the number of encoding bits per symbol for this sequence goes to zero. as $n \to \infty$.

   (c) Give a sequence for which the number of phrases in the LZ78 parsing grows as fast as possible.

   (d) Give a sequence for which the number of phrases in the LZ78 parsing grows as slowly as possible.