Homework Set #4

Due 27 October 2009, 6 pm, in INR036

Problem 1 (Simple optimum compression of a markov source)

Consider the three-state Markov process $U_1, U_2, \cdots$ having transition matrix given below.

$$\begin{array}{c|ccc} U_n & S_1 & S_2 & S_3 \\ U_{n-1} & & & \\ \hline S_1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ S_2 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ S_3 & 0 & \frac{1}{2} & \frac{1}{2} \\ \hline \end{array} \tag{1}$$

Thus the probability that S_1 follows S_3 is equal to zero. Design three codes C_1, C_2, C_3 (one for each state $1, 2$, and 3), each code mapping elements of the set of S_i's into sequences of 0's and 1's, such that this Markov process can be sent with maximal compression by the following scheme:

(a) Note the present symbol $X_n = i$.

(b) Select code C_i.

(c) Note the next symbol $X_{n+1} = j$ and send the codeword in C_i corresponding to j.

(d) Repeat for the next symbol. What is the average message length of the next symbol conditioned on the previous state $X_n = i$ using this coding scheme? What is the unconditional average number of bits per source symbol? Relate this to the entropy rate $H(\mathcal{U})$ of the Markov chain.

Problem 2 (Describing Types)

Define the *type* $P_{\mathbf{x}}$ (or empirical probability distribution) of a sequence $x_1, \cdots, x_n$ be the relative proportion of occurences of each symbol $\mathcal{X}$; i.e., $P_{\mathbf{x}}(a) = N(a|\mathbf{x})/n$ for all $a \in \mathcal{X}$, where $N(a|\mathbf{x})$ is the number of times the symbol a ocurs in the sequence $\mathbf{x} \in \mathcal{X}^n$.

(a) Show that if $X_1, \cdots X_n$ are drawn *i.i.d.* according to $Q(x)$, the probability of $\mathbf{x}$ depends only on its type and is given by

$$Q^n(\mathbf{x}) = 2^{-n(H(P_{\mathbf{x}})+D(P_{\mathbf{x}}\|Q))}.$$

Hint: Start by showing the following:

$$\begin{aligned} Q^n(\mathbf{x}) &= \prod_{i=1}^{n} Q(x_i) \\ &= \prod_{a\in\mathcal{X}} Q(a)^{N(a|\mathbf{x})} \\ &= \prod_{a\in\mathcal{X}} Q(a)^{nP_{\mathbf{x}}(a)} \end{aligned}$$

Define the *type class* $T(P)$ as the set of sequences of length n and type P:

$$T(P) = \{\mathbf{x} \in \mathcal{X}^n : \ P_{\mathbf{x}} = P\}.$$

For example, if we consider binary alphabet, the type is defined by the number of 1's in the sequence and the size of the type class is therefore $\binom{n}{k}$.

(b) It can be shown that

$$|T(P)| \doteq 2^{nH(P)}.$$

Prove this for binary alphabet by proving

$$\frac{1}{n+1} 2^{nH(\frac{k}{n})} \leq \binom{n}{k} \leq 2^{nH(\frac{k}{n})}.$$

Hint: To derive the upper bound start by proving

$$\begin{aligned} 1 &\geq \binom{n}{k} \left(\frac{k}{n}\right)^k \left(1 - \frac{k}{n}\right)^{n-k} \\ &= \binom{n}{k} 2^{n\left(\frac{k}{n}\log\frac{k}{n} + \frac{n-k}{n}\log\frac{n-k}{n}\right)} \end{aligned}$$

To derive the lower bound, start by proving the following chain of inequalities

$$\begin{aligned} 1 &= \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \\ &\leq (n+1) \max_k \binom{n}{k} p^k (1-p)^{n-k} \\ &= (n+1) \max_k \binom{n}{np} p^{np} (1-p)^{n-np}. \end{aligned}$$

(c) Use (a) and (b) to show that

$$Q^n(T(P)) \doteq 2^{-nD(P\|Q)}.$$

Problem 3 (Arithmetic Coding)

Let X_i be binary stationary Markov with transition matrix $\begin{pmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$.

(a) Find $F(01110) = Pr\{.X_1 X_2 \cdots X_5 < .01110\}$.

(b) How many bits $.F_1 F_2 \cdots$ can be known for sure if it is not known how 01110 continues?

Problem 4 (Lempel-Ziv-I)

Give the parsing and encoding of 00000011010100000110101 using the tree-structured Lempel-Ziv algorithm

Problem 5 (Lempel-Ziv-II)

In the siliding window variant of Lempel-Ziv, a short match can be represented by either (F, P, L) or (F, C), where F denotes the flag, P the pointer, L the lengh of the match, and C the uncompressed charater. Assume that the window length is W, and assume that the maximum match length is M.

(a) How many bits are required to represent P? to represent L?

(b) Assume that C, the representation of a character, is 8 bits long. As a function of W and M, what is the shortest match that one should represent as a match rather than as uncompressed characters?