

Homework Set #7

1. **Minimax regret data compression and channel capacity.** First consider universal data compression with respect to two source distributions. Let the alphabet $V = \{1, e, 0\}$ and let $p_1(v)$ put mass $1 - \alpha$ on $v = 1$ and mass α on $v = e$. Let $p_2(v)$ put mass $1 - \alpha$ on 0 and mass α on $v = e$.

We assign word lengths to V according to $l(v) = \log \frac{1}{p(v)}$, the ideal codeword length with respect to a cleverly chosen probability mass function $p(v)$. The worst case excess description length (above the entropy of the true distribution) is

$$\max_i \left(E_{p_i} \log \frac{1}{p(V)} - E_{p_i} \log \frac{1}{p_i(V)} \right) = \max_i D(p_i \parallel p). \quad (1)$$

Thus the minimax regret is $R^* = \min_p \max_i D(p_i \parallel p)$.

- (a) Find R^* .
- (b) Find the $p(v)$ achieving R^* .
- (c) Compare R^* to the capacity of the binary erasure channel

$$\begin{bmatrix} 1 - \alpha & \alpha & 0 \\ 0 & \alpha & 1 - \alpha \end{bmatrix}$$

and comment.

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

- (a) Find $F(01110) = \Pr\{.X_1X_2X_3X_4X_5 < .01110\}$.
- (b) How many bits $.F_1F_2 \dots$ can be known for sure if it is not known how $X = 01110$ continues?

3. **Lempel-Ziv.**

- (a) Continue the Lempel-Ziv parsing of the sequence 0,00,001,00000011010111.
- (b) Give a sequence for which the number of phrases in the LZ parsing grows as fast as possible.
- (c) Give a sequence for which the number of phrases in the LZ parsing grows as slowly as possible.

4. **Another idealized version of Lempel-Ziv coding.** An idealized version of LZ was shown to be optimal: The encoder and decoder both have available to them the “infinite past” generated by the process, \dots, X_{-1}, X_0 , and the encoder describes the string (X_1, X_2, \dots, X_n) by telling the decoder the position R_n in the past of the first recurrence of that string. This takes roughly $\log R_n + 2 \log \log R_n$ bits.

Now consider the following variant: Instead of describing R_n , the encoder describes R_{n-1} plus the last symbol X_n . From these two the decoder can reconstruct the string (X_1, X_2, \dots, X_n) .

- (a) What is the number of bits per symbol used in this case to encode (X_1, X_2, \dots, X_n) ?
- (b) Modify the proof given in the text to show that this version is also asymptotically optimal, namely that the expected number of bits-per-symbol converges to the entropy rate.

5. **Tunstall Coding:** The normal setting for source coding maps a symbol (or a block of symbols) from a finite alphabet onto a variable length string. An example of such a code is the Huffman code, which is the optimal (minimal expected length) mapping from a set of symbols to a prefix free set of codewords. Now consider the dual problem of variable-to-fixed length codes, where we map a variable length sequence of source symbols into a fixed length binary (or D -ary) representation. A variable-to-fixed length code for an i.i.d. sequence of random variables $X_1, X_2, \dots, X_n, X_i \sim p(x), x \in \mathcal{X} = \{0, 1, \dots, m-1\}$ is defined by a prefix-free set of phrases $A_D \subset \mathcal{X}^*$, where \mathcal{X}^* is the set of finite length strings of symbols of \mathcal{X} , and $|A_D| = D$. Given any sequence X_1, X_2, \dots, X_n , the string is parsed into phrases from A_D (unique because of the prefix free property of A_D), and represented by a sequence of symbols from a D -ary alphabet. Define the efficiency of this coding scheme by

$$R(A_D) = \frac{\log D}{EL(A_D)} \quad (2)$$

where $EL(A_D)$ is the expected length of a phrase from A_D .

Prove that $R(A_D) \geq H(X)$.

6. **Optimal Integer Codes.** Consider a variation of the simple integer encoding described in Lemma 13.5.1. To encode the integer k : represent $\lceil \log(k+1) \rceil$ in unary, followed by the binary representation of k using $\lceil \log(k+1) \rceil$ bits, the first bit of which is 1, i.e.,

$$C_1(k) = \underbrace{00\dots0}_{\lceil \log(k+1) \rceil \text{ 0's}} \underbrace{1xx\dots x}_{k \text{ in binary}} \quad (3)$$

It is easy to see that the length of this representation is $2\lceil \log(k+1) \rceil$.

- (a) Argue that this representation is prefix free.
- (b) Assume that the codeword 1 is used to represent the symbol “0”, and the other codewords are used to represent the integers $1, 2, \dots$. For what distribution on the integers is this encoding optimal?