

Information Theory with Applications

MATH 6397 – Fall 2008

October 13, 2008

Homework Set 2, due Thu Oct 16, 2008

Unless noted, exercises are taken from the textbook. Additional hints may be available there.

1. **p. 115 Ex. 3.4** Find an example of a regular (invertible) K -ary variable length code for a discrete memoryless source $\{X_j\}$ with marginal Q on a finite non-empty alphabet \mathbb{A} such that $\mathbb{E}[\ell(X_1)] < H_K(Q)$.
2. **p. 117 Ex. 3.10** Let \mathcal{T} be the code tree for a prefix code which encodes a discrete memoryless source with marginal Q on a finite alphabet \mathbb{A} . Label the nodes in the tree by the sum of the probabilities of the descendant leaves, as discussed in class. Prove the following:

Denumerate the nodes in an arbitrary fashion. Let the children of node j be i_1, i_2, \dots, i_k and their respective probabilities $q_{i_1}, q_{i_2}, \dots, q_{i_k}$. If $q_j \neq 0$, define the conditional probability measure Q_j supported on the children by normalizing their probabilities, $q'_{i_1} = q_{i_1}/q_j, q'_{i_2} = q_{i_2}/q_j$, etc., then

$$H(Q) = \sum_{j=1}^{\text{\#nodes}} q_j H(Q_j).$$

3. **p. 117 Ex. 3.11** With the same notation as in the preceding exercise, show that if \mathcal{T} is an optimal code tree (average code length assumes the minimum value) then for any two nodes i and j and their levels (root has level zero) l_i and l_j ,

$$-1 - \frac{q_i}{q_j} \leq l_i - l_j \leq 1 + \frac{q_j}{q_i}.$$

4. **p. 124 Ex. 3.28** Generalize the construction of an optimal binary prefix code according to Huffman to the countably infinite alphabet $\mathbb{A} = \mathbb{N}$, assuming that the marginal Q of the discrete memoryless source $\{X_j\}$ satisfies $q_1 = Q(X_1 = 1) \geq q_2 = Q(X_1 = 2) \geq q_3 = Q(X_1 = 3) \geq \dots$ and for infinitely many $m \in \mathbb{N}$,

$$Q(X_j = m) \geq Q(X_j \geq m + 1).$$

Hint: Given the sequence $\{m_j\}_{j=1}^{\infty}$ consider first a Huffman code tree \mathcal{H}_1 for the alphabet $\mathbb{A}_1 = \{1, 2, \dots, m + 1\}$ with probabilities q_1, q_2, \dots, q_m , and $q'_{m+1} = \sum_{j \geq m+1} q_j$. Now grow the tree inductively while preserving optimality.

5. **Matlab project** Design an algorithm that builds a prefix code for the text at www.math.uh.edu/~bgb/Courses/Math6397-F08/dq.txt Try to achieve the smallest average code length.

Your project includes generating three matlab scripts (or functions). The first one, `codegen.m` builds a code which is suitable for the relative frequencies extracted from the text. This fixed-variable code is intended to parse text in blocks of 4 characters (including spaces). To define a prefix code, denumerate the 27^4 different sequences of length 4 containing 26 letters and space in lexicographic order. Now assign code words by creating a cell array C of strings containing “0”s and “1”s such that the j -th 4-letter word in the alphabet is mapped to the string $C\{j\}$.

The second matlab `encode.m` script uses the generated codebook on a benchmark passage (1-2 pages) of the text at www.math.uh.edu/~bgb/Courses/Math6397-F08/dq_wind.txt. The output of `encode.m` is supposed to be a string of “0” and “1”s which are concatenated codewords. Feel free to test your code beforehand with passages of your liking.

As part of this script, compute the compression rate (length of output string * log 2 / length of input string * log 27) and document the output string length and compression ratio when `encode.m` is applied to the benchmark passage.

The third matlab script `decode.m` restores the un-encoded input passage from the encoded string. Compare your output with the un-encoded input passage to make sure your code works.

Print and attach your matlab scripts and the results (compression ratio and decoded text) for the benchmark passage.