Information Theory with Applications

MATH 6397 – Fall 2014

Homework Set 2, due Thursday, Oct 9, 2014

1. Let $\mathcal{T}$ be the code tree for a prefix code which encodes a discrete memoryless source with induced probability measure $\mathbb{Q}$ on a finite alphabet $\mathbb{A}$. Label the nodes in the tree by the sum of the probabilities of the descendent 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 $\mathbb{Q}_j$ supported on the children $\{i_1, i_2, \dots i_k\}$ by normalizing their probabilities, $\mathbb{Q}_j(i_1) = q_{i_1}/q_j$, $\mathbb{Q}_j(i_2) = q_{i_2}/q_j$, etc., then
   >
   > $$H(\mathbb{Q}) = \sum_{j=1}^{\#\text{nodes}} q_j H(\mathbb{Q}_j)\,.$$

   Hint: Perform induction over the number of nodes in a tree.

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

   $$\mathbb{Q}(X_j = m) \geq \mathbb{Q}(X_j \geq m+1)\,.$$

   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.

3. **Matlab project** Design an algorithm that builds a prefix code for the text at www.math.uh.edu/∼bgb/Courses/Math6397F14/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 script `encode.m` uses the generated codebook on a benchmark passage (1-2 pages) of the text at www.math.uh.edu/∼bgb/Courses/Math6397F14/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.