CSC 310, Spring 2004 — Assignment #1 Solutions

Question 1 (15 marks):

a) Explain why the code with codewords $\{0,\ 0010,\ 0001100\}$ is uniquely decodable. A convincing informal explanation is OK.

This code can be decoded by looking for ocurrences of the substrings 010 *and* 0110*. The first can occur only as part of the codeword* 0010*, the second only as part of the codeword* 0001100*. After finding these substrings, mark the bits that make up the corresponding codewords. (Note that the occurrences of* 010 *and* 0110 *cannot overlap, since clearly two consecutive 1s will be separated by at least three 0s in a string produced by this code.) The remaining unmarked bits will all be 0s, and each of those bits is an instance of the first codeword,* 0*.*

b) Is the code in part (a) instantaneous? Show why or why not.

No, since the codeword 0 *is a prefix of the codeword* 0010 *(and also of the codeword* 0001100*).*

c) Show that the code with codewords $\{0,\ 0010,\ 000100\}$ is not uniquely decodable, by giving an example of a string of bits that can be decoded in more than one way.

The string 000100 *can be decoded as the single codeword* 000100 *or as the three codewords* 0, 0010, *and* 0*.*

d) Does the code in part (c) satisfy the Kraft-McMillan inequality?

Yes, since $1/2^1 + 1/2^4 + 1/2^6 = 37/64$*, which is less than or equal to one. Note: The fact that a code satisfies the Kraft-McMillian inequality does not guarantee that it is uniquely decodable.*

Question 2 (25 marks): Consider a source with source alphabet $\{a_1, a_2, a_3, a_4, a_5, a_6\}$ in which the symbols probabilities are as follows:

$$p_1 = 0.27,\ p_2 = 0.09,\ p_3 = 0.23,\ p_4 = 0.11,\ p_5 = 0.15,\ p_6 = 0.15$$

a) Compute the entropy of this source.

The entropy is

$$\begin{aligned} &0.27\log_2(1/0.27) \ + \ 0.09\log_2(1/0.09) \ + \ 0.23\log_2(1/0.23) \\ &\quad + \ 0.11\log(1/0.11) \ + \ 0.15\log(1/0.15) \ + \ 0.15\log(1/0.15) \ = \ 2.4817 \end{aligned}$$

Note that $\log_2(x) = \log_e(x)/\log_e(2)$.

b) Find a Huffman code for this source. Show your work.

The two least probable symbols are a_2 *and* a_4*. They will be merged to a new combined symbol with probability 0.20. The next two least probable symbols are* a_5 *and* a_6*. They will be merged to a new combined symbol with probability 0.30. The first of these merged symbols will then be merged with* a_3 *and the second of the merged symbols will be merged with* a_1*, and finally these two new merged symbols will be merged with each other.*

One possible Hufman code, obtained if the first symbol mentioned above for each merge is assigned to the 0 bit and the second to the 1 bit, is

$$\begin{aligned} a_1 &: 11 \\ a_2 &: 000 \\ a_3 &: 01 \\ a_4 &: 001 \\ a_5 &: 100 \\ a_6 &: 101 \end{aligned}$$

c) Compute the expected codeword length for the Huffman code you found in part (b).

$0.27 \times 2 \;+\; 0.09 \times 3 \;+\; 0.23 \times 2 \;+\; 0.11 \times 3 \;+\; 0.15 \times 3 \;+\; 0.15 \times 3 \;=\; 2.5$

d) Find an optimal instantaneous code for this source that is *not* a Huffman code. (Your code should not just be different from the Huffman code you found in part (b); it must also be different from any other Huffman code that could be obtained by changing the way arbitrary choices are made in the Huffman code algorithm.)

We can get another optimal code from this Huffman code by swapping the assignment of codewords to symbols for two symbols whose codewords are of equal length. For example, swapping the codewords for a_4 and a_5 produces the following code, which is uniquely decodable and also has minimum expected codeword length:

$$\begin{aligned} a_1 &: 11 \\ a_2 &: 000 \\ a_3 &: 01 \\ a_4 &: 100 \\ a_5 &: 001 \\ a_6 &: 101 \end{aligned}$$

It is uniquely decodable because the set of codewords is identical to the Huffman code above. It is optimal because every symbol has a codeword of the same length as the Huffman code. But it cannot be a Huffman code, because the two symbols with smallest probability, a_2 and a_4, have codewords that differ in more than the last bit (whereas in any Huffman code, these codewords will be "siblings" in the tree).

Question 3 (30 marks): Let C be a uniquely-decodable binary code for a source with symbol probabilities $p_1, \ldots, p_I$ in which the codewords for these symbols have lengths $l_1, \ldots, l_I$. Suppose that for some distinct i, j, and k, $l_i = l_j = l_k$. Prove that if C is optimal (ie, has minimal expected codeword length), then $p_i \leq p_j + p_k$. Caution: C is not necessarily a Huffman code; it might not even be instantaneous.

First proof: *We will suppose that $p_i > p_j + p_k$ and derive a contradiction. Since C is uniquely decodable, it satisfies the Kraft-McMillan inequality, which can be written as*

$$\frac{1}{2^{l_i}} + \frac{1}{2^{l_j}} + \frac{1}{2^{l_k}} + \sum_{h \notin \{i,j,k\}} \frac{1}{2^{l_h}} \leq 1$$

From the fact that $l_i = l_j = l_k$, we can conclude that

$$\frac{1}{2^{l_i}} + \frac{1}{2^{l_j}} + \frac{1}{2^{l_k}} = \frac{1}{2^{l_i - 1}} + \frac{1}{2^{l_j + 1}} + \frac{1}{2^{l_k + 1}}$$

It follows that

$$\frac{1}{2^{l_i-1}} + \frac{1}{2^{l_j+1}} + \frac{1}{2^{l_k+1}} + \sum_{h\notin\{i,j,k\}} \frac{1}{2^{l_h}} \le 1$$

from which we can conclude that there exists a uniquely-decodable code, C', in which symbol a_i has a codeword of length $l'_i = l_i - 1$, symbols a_j and a_k have codewords of lengths $l'_j = l_j + 1$ and $l'_k = l_k + 1$, and all other symbols have codewords that are the same lengths as in C.

The expected codeword length for C' is

$$\begin{aligned} & p_i(l_i-1) + p_j(l_j+1) + p_k(l_k+1) + \sum_{h\notin\{i,j,k\}} p_h l_h \\ = \ & p_i l_i + p_j l_j + p_k l_k + \sum_{h\notin\{i,j,k\}} p_h l_h - (p_i - p_j - p_k) \\ < \ & p_i l_i + p_j l_j + p_k l_k + \sum_{h\notin\{i,j,k\}} p_h l_h \end{aligned}$$

The last inequality comes about because $p_i - p_j - p_k$ is positive, since we have assumed that $p_i > p_j + p_k$. The right side of this inequality is the expected codeword length for the original code, C. But if C' has smaller expected codeword length than C, C cannot be an optimal code. So we see that assuming $p_i > p_j + p_k$ leads to a contradiction. This assumption must therefore be wrong, and hence $p_i \le p_j + p_k$.

Second proof: *We will suppose that $p_i > p_j + p_k$ and derive a contradiction. Since C is uniquely decodable, the lengths of its codewords must satisfy the Kraft-McMillan inequality. Since these lengths satisfy the Kraft-McMillan inequality, there exists an instantaneous code with these same codeword lengths. Let C' be such an instantaneous code.*

Since C' is instantaneous, it can be represented as a tree, with codewords at the leaves. Consider the leaves corresponding to the codewords for symbols a_i, a_j, and a_k, all of which will be at the same level in the tree. If the codewords for a_j and a_k are not siblings, swap the codeword for a_j and the subtree that is the sibling of a_k to produce another instantaneous code, C'', with the same codeword lengths as C' (and C), but in which the codewords for a_j and a_k are siblings.

The codeword for a_i in C'' can be written as xb, where x is some string of bits and b is a single bit. The codewords for a_j and a_k will have the forms $y0$ and $y1$, where y is some string of bits, which will be the same length as x. We now create a new code, C''', in which the codeword for a_i is y, and the codewords for a_j and a_k are $xb0$ and $xb1$, with the codewords for all other symbols being the same as in C''. (This change is most easily visualized by drawing the code trees.) It is easy to see that C''' is an instantaneous code. Since C''' encodes a_1 using one less bit than C'', and a_2 and a_3 using one more bit than C'', the difference of the expected codeword length of C''' and that of C'' is $-p_1 + p_j + p_k$. If $p_1 > p_j + p_k$, this difference is negative, which is impossible if C'' is optimal. Since assuming $p_i > p_j + p_k$ leads to a contradiction, this assumption must be wrong, and hence $p_i \le p_j + p_k$.

Question 4 (30 marks total): Suppose a source produces independent symbols from the alphabet $\{a_1,\ a_2,\ a_3\}$, with probabilities $p_1 = 0.4999999$, $p_2 = 0.4999999$, and $p_3 = 0.0000002$.

a) Compute the entropy of this source.

The entropy is

$$-0.4999999\log_2(0.4999999) - 0.4999999\log_2(0.4999999) - 0.0000002\log_2(0.0000002)$$
$$= 1.000004539$$

b) Find an optimal code for this source, and compute its expected codeword length.

One optimal code is the following:

$$\begin{array}{lcl} a_1 & : & 0 \\ a_2 & : & 10 \\ a_3 & : & 11 \end{array}$$

Its expected codeword length is

$$0.4999999 \times 1 \ + \ 0.4999999 \times 2 \ + \ 0.0000002 \times 2 \ = \ 1.5000001$$

c) Find an optimal code for the second extension of this source (ie, for blocks of two symbols), and compute its expected codeword length, and the expected codeword length divided by two.

Here is one optimal code:

$$\begin{array}{lcl} (a_1, a_1) & : & 00 \\ (a_1, a_2) & : & 01 \\ (a_2, a_1) & : & 10 \\ (a_2, a_2) & : & 110 \\ (a_1, a_3) & : & 11100 \\ (a_2, a_3) & : & 11101 \\ (a_3, a_1) & : & 11110 \\ (a_3, a_2) & : & 111110 \\ (a_3, a_3) & : & 111111 \end{array}$$

Its expected codeword length is approximately 2.2500012, which divided by two is approximately 1.1250006.

d) Prove (without any tedious calculations) that in order to compress to within 1% of the entropy by encoding blocks of size N from this source, N will have to be at least 5.

When N is less than 5, the 2^N blocks in which all symbols are either a_1 or a_2 will be much more probable than all other blocks. An optimal code for the N-th extension will assign codewords of length N to all but one of these high-probability blocks, and a codeword of length $N+1$ to the remaining high-probability block. (Other blocks will have codewords of length greater than $N+1$. The expected codeword length for such a code will be greater than

$$N \times (2^N - 1) \times 0.4999999^N + (N+1) \times 0.4999999^N \ = \ N \times 2^N \times 0.4999999^N + 0.4999999^N$$

The expected codeword length divided by N will be greater than $2^N \times 0.4999999^N + 0.4999999^N/N$. For $N = 1$, $N = 2$, $N = 3$, and $N = 4$, the values of this expression are 1.4999997, 1.1249996, 1.0416660, and 1.0156242. The entropy plus 1% is 1.0100046. Since the expected codeword length divided by N is greater than this for $N < 5$, a block size of at least five will be needed to compress to within 1% of the entropy.