Model Answers to Exercise 5 of October 15, 2014

Problem 1  

Huffman Coding

(a) The Huffman tree for this distribution is

(b) The expected length of the codewords for the binary Huffman code is

\[
L(C) = E[l(X)] = 0.49 \cdot 1 + 0.26 \cdot 2 + 0.12 \cdot 3 + 0.04 \cdot 5 + 0.04 \cdot 5 + 0.03 \cdot 5 + 0.02 \cdot 5 \\
= 2.02 \text{ bits.}
\]

Note that \( H(X) \approx 2.013 \) bits.

(c) The ternary Huffman tree is

This code has an expected length

\[
L(C) = E[l(X)] = 0.49 \cdot 1 + 0.26 \cdot 1 + 0.12 \cdot 2 + 0.04 \cdot 2 + 0.04 \cdot 3 + 0.03 \cdot 3 + 0.02 \cdot 3 \\
= 1.34 \text{ ternary symbols.}
\]

Note that \( H_3(X) \approx 1.270 \) ternary symbols.
Problem 2

Bad Codes

a) \{0, 10, 11\} is a Huffman code, e.g., for the distribution \(\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)\).

b) The code \{00, 01, 10, 110\} can be shortened to \{00, 01, 10, 11\} without losing its instantaneous property, and is therefore not optimal. Thus, it cannot be a Huffman code. Alternatively, it is not a Huffman code because there is a unique longest codeword.

c) The code \{01, 10\} can be shortened to \{0, 1\} without losing its instantaneous property, and is therefore not optimal and, consequently, not a Huffman code.

Problem 3

Optimal Codeword Lengths

Because of the assumption
\[ p_1 > p_2 \geq p_3 \geq \ldots \geq p_m \quad (1) \]

we can without loss of generality assume
\[ l_1 \leq l_2 \leq \ldots \leq l_m. \quad \text{ (2) } \]

a) We prove by contradiction that for
\[ p_1 > \frac{2}{5} \quad (2) \]

the most probable codeword \(x_1\) must have length \(l_1 = 1\).

Suppose that there exists a Huffman code with \(l_1 = 2\). Then, the corresponding tree looks as follows:

\[
\begin{align*}
1.0 & \quad p_1 \\
& \quad a \quad q_a \\
& \quad b \quad q_b \\
& \quad q_b
\end{align*}
\]

where node a (which may but need not be a codeword) with probability \(q_a\) must exist because otherwise the corresponding code would be strictly suboptimal and hence not a Huffman code. Thus, we know
\[ p_1 + q_a + q_b = 1, \quad (3) \]
\[ q_a \leq q_b, \quad (4) \]
\[ p_1 \leq q_b, \quad (5) \]

where (3) follows from the properties of probability mass functions, and (4) and (5) hold because otherwise the code would not be optimal. (Remember that the code under consideration has to be optimal since it is by assumption a Huffman code.) Equations (5) and (2) imply \(q_b > \frac{2}{5}\). Moreover, because of (1), node b with probability \(q_b\) cannot be a codeword. Thus, the tree has to look as follows:
Now, we can consider the construction algorithm of the Huffman code. If node a and $p_1$ are combined before nodes c and d, then we know that

$$q_c \geq \max\{p_1, q_a\} > \frac{2}{5},$$

$$q_d \geq \max\{p_1, q_a\} > \frac{2}{5}.$$

These two conditions and (2) imply $p_1 + q_c + q_d > \frac{6}{5}$, which contradicts (3). Thus, nodes c and d are combined first. Equations (3), (2), and (5) imply $q_a < \frac{1}{5}$. We therefore have

$$q_c + q_d = q_b > \frac{2}{5},$$

$$q_c \leq \min\{p_1, q_a\} < \frac{1}{5},$$

$$q_d \leq \min\{p_1, q_a\} < \frac{1}{5},$$

which contradicts $q_b > \frac{2}{5}$. We conclude that $\ell(x_1) = 2$ must be wrong.

As we argue next, the assumption $l_1 > 2$, leads to a contradiction because of the above analysis for the case $l_1 = 2$. To see this, consider the node on the shortest path between the node corresponding to the probability $p_1$ and the root that is separated from the root by two edges. Observe that the probability $q$, which corresponds to this node, has to be greater than $p_1$. Hence, we obtain a contradiction from the analysis for the case $l_1 = 2$ and we therefore conclude that the length of the codeword for $x_1$ must be 1.

A shorter but less instructive proof can be obtained by induction over $m$. Since the statement is trivially true for $m = 2$, we assume without loss of generality $m \geq 3$. Indeed, we obtain from the inductive nature of the Huffman algorithm:

**Basis:** If $m = 3$, then the statement $l_1 = 1$ is trivially true since the Huffman algorithm allocates length 2 only to the two least likely symbols.

**Inductive Step:** Suppose the statement is true for $m = k \geq 3$ and consider $m = k+1$. In the first step, the Huffman procedure combines two least likely symbols of probabilities $p_m$ and $p_{m-1}$. If $p_m + p_{m-1} > \frac{2}{5}$, then $p_2, \ldots, p_{m-1} > \frac{1}{5}$ and hence $\frac{2}{5} < p_1 = 1 - p_2 - \ldots - p_m < 1 - (m-3)\frac{1}{5} - \frac{2}{5} \leq \frac{2}{5}$, which is a contradiction. Hence, $p_m + p_{m-1} \leq \frac{2}{5}$ and we can evoke the induction hypothesis to conclude that $l_1 = 1$.

b) We prove by contradiction that for

$$p_1 < \frac{1}{3}$$

the most probable codeword $x_1$ must have length $l_1 \geq 2$.

For contradiction, suppose there exists a Huffman code with $l_1 = 1$. Then the corresponding tree looks as follows:
Thus, we know
\[ p_1 + q_a = 1. \]

The last relation and (6) imply that \( q_a > \frac{2}{3} \). Because of (1) \( p_1 \) is the largest probability and node \( a \) therefore cannot be a codeword. Thus the tree has to look as follows:

Furthermore,
\[ q_b + q_c = q_a > \frac{2}{3}, \quad (7) \]
\[ q_b \leq p_1, \quad (8) \]
\[ q_c \leq p_1, \quad (9) \]

where (8) and (9) hold since otherwise in the construction of the Huffman code one would have combined either node \( b \) or node \( c \) with the node that corresponds to \( x_1 \) instead of combining the nodes \( b \) and \( c \). From (8) and (9) we obtain
\[ q_b + q_c \leq 2 \cdot p_1 < \frac{2}{3} \quad (10) \]

and hence a contradiction to (7). Thus, the assumption that \( l_1 = 1 \) must be wrong.

Problem 4 \textit{AEP}

a) From the weak law of large numbers, we know that for IID random variables \( Y_i \), we have
\[ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} Y_i = \mathbb{E}[Y_i] \]

in probability. We use the fact that the random variables \( X_1, X_2, \ldots \) are IID (and hence \( q(X_1), q(X_2), \ldots \) are also IID) to obtain
\[ \lim_{n \to \infty} \left( -\frac{1}{n} \log q(X_1, \ldots, X_n) \right) = \lim_{n \to \infty} \left( -\frac{1}{n} \log \prod_{i=1}^{n} q(X_i) \right) = \lim_{n \to \infty} \left( -\frac{1}{n} \sum_{i=1}^{n} \log q(X_i) \right) \]
\[= -\mathbb{E} [\log q(X)] \quad \text{in probability} \]
\[= - \sum_{x=1}^{m} p(x) \log q(x) \]
\[= \sum_{x=1}^{m} p(x) \log \frac{p(x)}{q(x)} - \sum_{x=1}^{m} p(x) \log p(x) \]
\[= D(p||q) + H(p). \]

b) Using the AEP and the result in a) we get
\[\lim_{n \to \infty} \frac{1}{n} \log \frac{p(X_1, \ldots, X_n)}{q(X_1, \ldots, X_n)} = \lim_{n \to \infty} \left( \frac{1}{n} \log p(X_1, \ldots, X_n) - \frac{1}{n} \log q(X_1, \ldots, X_n) \right) \]
\[= -H(p) + H(p) + D(p||q) \quad \text{in probability} \]
\[= D(p||q). \]

Problem 5

**Proof of Theorem 3.3.1 in Cover & Thomas: High-Probability Sets and the Typical Set**

Without loss of generality we will concentrate on the logarithm to the base 2 and the entropy given in bits.

a) Using
\[\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)\]

we get
\[\Pr(A \cap B) = \Pr(A) + \Pr(B) - \Pr(A \cup B) \]
\[> 1 - \epsilon_1 \]
\[> 1 - \epsilon_2 \]
\[\leq 1 \]
\[= 1 - \epsilon_1 - \epsilon_2. \]

We now choose \( A \triangleq A^{(n)}_\epsilon \) and \( B \triangleq B^{(n)}_\delta \) such that \( \epsilon_1 = \epsilon, \epsilon_2 = \delta \). Then from above it immediately follows that
\[\Pr\left( A^{(n)}_\epsilon \cap B^{(n)}_\delta \right) > 1 - \epsilon - \delta. \]

b) The first inequality follows from Part a); the subsequent equality from the definitions of the sets \( A^{(n)}_\epsilon \) and \( B^{(n)}_\delta \) as being sets of sequences \( x \in X^n \); the subsequent inequality from the fact that since \( x \in A^{(n)}_\epsilon \cap B^{(n)}_\delta \) the sequence \( x \) must be element of \( A^{(n)}_\epsilon \) and from the property of any element of \( A^{(n)}_\epsilon \)
\[P_X(x) \leq 2^{-n(H(X) - \epsilon)}; \]

the subsequent equality follows from counting the number of summands in the sum; and the final inequality by the fact that the number of elements in \( A \cap B \) cannot be larger than the number of elements in \( B \).

c) From Part b) we have
\[|B^{(n)}_\delta| > (1 - \epsilon - \delta) 2^{n(H(X) - \epsilon)} \]

Since it was assumed that \( \epsilon < \frac{1}{2} \) and \( \delta < \frac{1}{2} \) it follows that \( 1 - \epsilon - \delta > 0 \) and we can take the logarithm on both sides:
\[\log_2 |B^{(n)}_\delta| > \log_2 (1 - \epsilon - \delta) + n(H(X) - \epsilon) \]

© Amos Lapidoth, 2014/2015 5
from which follows
\[ \frac{1}{n} \log_2 |B_\delta^{(n)}| > H(X) - \delta' \]
where \( \delta' = \epsilon - \frac{1}{n} \log_2 (1 - \epsilon - \delta) \) which can be made arbitrarily small by an appropriate choice of \( \epsilon \) and \( \delta \) followed by an appropriate choice of \( n \).

**Problem 6**

*From AEP to Kraft’s Inequality*

Assume that there exists a prefix-free one-to-variable code \( C \) whose codeword lengths \( \ell_1, \ldots, \ell_d \) satisfy
\[ \sum_{i=1}^{d} 2^{-\ell_i} = \alpha > 1. \] (11)

We construct a source \( X \) with \( d \) outcomes where the \( i \)th outcome has probability
\[ p_i = \frac{2^{-\ell_i}}{\alpha}. \]

Using \( C \) to encode \( X \) yields the expected codeword length
\[ \mathbb{E}[\ell(X)] = \sum_{i=1}^{d} \frac{2^{-\ell_i}}{\alpha} \cdot \ell_i = \sum_{i=1}^{d} p_i \log \frac{1}{\alpha p_i} = H(X) - \log \alpha. \]

Fix \( \delta \in (0, \log \alpha) \). For every \( n \in \mathbb{Z}^+ \) we construct a fixed-to-fixed code using \( \lceil n(H(X) - \log \alpha + \delta) \rceil \) bits as follows:

**Step 1:** Describe every source sequence \( \mathbf{x} \in \mathcal{X}^n \) using the extension of \( C \).

**Step 2:** If the length of the description of \( \mathbf{x} \) is less than or equal to \( \lceil n(H(X) - \log \alpha + \delta) \rceil \), add zeros to its end to extend its length to \( \lceil n(H(X) - \log \alpha + \delta) \rceil \) and use it as the codeword for \( \mathbf{x} \).

**Step 3:** If the length of the description of \( \mathbf{x} \) is larger than \( \lceil n(H(X) - \log \alpha + \delta) \rceil \), use the all-zero bitstring as the codeword for \( \mathbf{x} \).

First, observe that the rate of the described code satisfies
\[ \lim_{n \to \infty} p_n = \lim_{n \to \infty} \frac{n(H(X) - \log \alpha + \delta)}{n} = H(X) - \log \alpha + \delta < H(X). \] (12)

We next analyze the error probability of the code. Note that an error occurs if, and only if, the length of the description of \( \mathbf{x} \) in Step 1 is larger than \( \lceil n(H(X) - \log \alpha + \delta) \rceil \). This length can be written as
\[ \sum_{k=1}^{n} \ell(x_k). \]
By the Law of Large Numbers, as $n$ tends to infinity,

$$
\frac{1}{n} \sum_{k=1}^{n} \ell(X_k) \rightarrow \mathbb{E}[\ell(X)] = H(X) - \log \alpha \quad \text{in probability.}
$$

Therefore, as $n$ tends to infinity, the probability that

$$
\sum_{k=1}^{n} \ell(X_k) > \left\lceil n(H(X) - \log \alpha + \delta) \right\rceil
$$

tends to zero. Hence the probability of successful decoding for these codes tends to one as $n$ tends to infinity. This fact combined with (12) contradicts the theorem. Thus we conclude that there exists no prefix-free code whose codeword lengths satisfy (11).