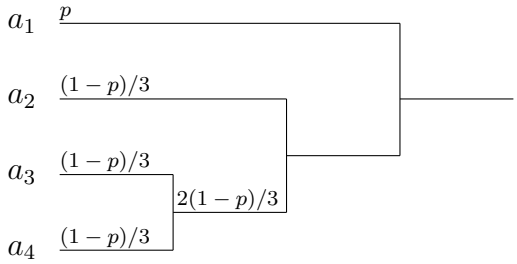
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

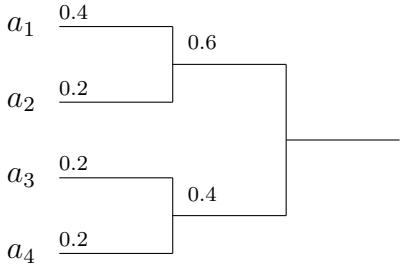
Handout 4 Information Theory and Coding
Solutions to homework 2 October 26, 2007

PROBLEM 1.

(a) Let $p = P(a_1)$, thus $P(a_2) = P(a_3) = P(a_4) = (1-p)/3$. By the Huffman construction (see figure below) we must have $p > 2(1-p)/3$, i.e., $q = 2/5$ in order to have $n_1 = 1$.



(b) With $P(a_1) = q$, the figure below illustrates that a Huffman code exists with $n_1 > 1$.



(c) & (d) For $K = 2$, n_1 is always 1. For $K = 3$, $n_1 = 1$ is guaranteed by $P(a_1) > P(a_2) \geq P(a_3)$. Now take $K \geq 4$ and assume $P(a_1) > 2/5$. This implies that

$$\sum_{i=2}^{K} P(a_i) < 3/5 \tag{1}$$

Also we have $P(a_1) > P(a_2) \geq \cdots \geq P(a_K)$. Now we claim that $P(a_{K-1}) + P(a_K) < 2\frac{3/5}{K-1}$. Indeed, if $K-1$ is even then we have that $P(a_i) + P(a_{i+1}) \geq P(a_K) + P(a_{K-1})$ for all $2 \leq i \leq K-3$. Thus we have from equation (1)

$$\frac{K-1}{2}(P(a_{K-1}) + P(a_K)) < 3/5$$

which implies that $(P(a_{K-1}) + P(a_K)) < \frac{2}{K-1}3/5$. If $K-1$ is odd then $K-2$ is even and we have $P(a_i) + P(a_{i+1}) \geq P(a_K) + P(a_{K-1})$ for all $3 \leq i \leq K-3$. Also $\frac{P(a_2)}{2} \geq \frac{P(a_K)}{2}$ and $\frac{P(a_2)}{2} \geq \frac{P(a_{K-1})}{2}$. Thus $P(a_2) \geq \frac{P(a_K + P(a_{K-1})}{2}$. Thus putting it all we get $\big(P(a_K + P(a_{K-1})\big)(\frac{K-2}{2} + \frac{1}{2}) < 3/5$ which implies that $\big(P(a_K + P(a_{K-1})\big)(\frac{K-1}{2}) < 3/5$. Thus proving the claim.

Now the Huffman procedure will combine a_{K-1} and a_K to obtain a super-symbol with probability

$$P(a_{K-1}) + P(a_K) < 2\frac{3/5}{K-1} \leq 2/5.$$

Thus, in the reduced ensemble a_1 is still the most likely element. Repeating the argument until $K = 3$, we see that $P(a_1) > q$ guarantees $n_1 = 1$ in all cases.

(e) For $K \leq 3$ no such q' exists. For $K > 3$, we claim $q' = 1/3$. Assume a_1 remains unpaired until the 2nd to last stage (otherwise there is nothing to prove). At this stage we have three nodes, and $P(a_1) < q'$ must be strictly less than one of the other two (otherwise all three would have been less than 1/3). Thus a_1 will be combined with one of them, leading to $n_1 > 1$.

PROBLEM 2.

It is clear that if $x = 1$ then $K = 2^j$ and the optimal code will assign length j for all codewords. If $x = 2$ (I know $x = 2$ is not allowed, but...) then $K = 2^{j+1}$ and the optimal code will assign length $j + 1$ to all symbols. Now we have $K = x2^j$ for $1 \leq x < 2$. So intuitively we feel that all codewords should be placed either at length j or at length $j + 1$. To prove this intuitive fact we prove that the length of the shortest codeword and the longest codeword can not be greater than 1 bit. Having proved this we see that the lengths cannot be $j - i$ and $j - i + 1$ for $i \geq 1$. This because the average length of the codewords is strictly less than j, whereas the entropy of the source is $j + \log_2 x$ which is $\geq j$, thus violating the fact that the average length of a code is greater than or equal to the entropy of the source. And if the lengths are $j + i$ and $j + i + 1$ for $i \geq 1$ then we can just shift the codewords up the tree. Hence the optimal code would have lengths j and $j + 1$.

(a) We prove here that the longest and the shortest codeword differ by at most one bit. Consider the longest and the shortest codewords. We know that there are at least two longest codewords, suppose their length is l. Suppose the shortest codewords has length s. If s and l differ by more than 1, then we can increase the length of the shortest codeword by 1 ($s' = s + 1$) and shorten the two longest codewords by 1 ($l' = l - 1$) and still satisfy Kraft inequality:

$$[2^{-s'} + 2^{-l'} + 2^{-l'}] - [2^{-s} + 2^{-l} + 2^{-l}] = 2^{-(l-1)} - 2^{-(s+1)} \leq 0.$$

But since all the codewords are equally likely, this would have decreased the average codeword length, contradicting the optimality of the Huffman code. Thus, the longest and shortest codeword lengths can differ by at most 1, and, again by Kraft inequality, their lengths must be j and $j + 1$.

(b) Let the number of codewords of length k be m_k, $k = j, j+1$. Since Huffman procedure yields a complete tree all intermediate nodes have two children. Thus, the 2^j nodes at level j of the tree are either codewords (m_j of them) or each of their two children are codewords ($m_{j+1}/2$ of them). Thus

$$m_j + m_{j+1}/2 = 2^j,$$

and also $m_j + m_{j+1} = x2^j$. From these two equations we find

$$m_j = (2 - x)2^j \quad \text{and} \quad m_{j+1} = (x - 1)2^{j+1}.$$

(c) By the result above the average codeword length is

$$[jm_j + (j+1)m_{j+1}]/(x2^j) = j + 2(x-1)/x.$$

PROBLEM 3.

(a) $\{00, 01, 100, 101, 1100, 1101, 1110, 1111\}$.

(b) For $i > j$ observe that

$$Q_i - Q_j = \sum_{k=j}^{i-1} P(a_k) \geq P(a_j) \geq 2^{-l_j}.$$

So, the binary expansion of Q_i and Q_j must differ somewhere in the first l_j bits (if they did not the difference between Q_i and Q_j would have been less than 2^{-l_j}). Since codewords for i and j are at least l_j bits long, this implies that neither codeword can be a prefix of the other. The bound on the average codeword length follows from

$$-\log_2 P(a_i) \leq l_i < -\log_2 P(a_i) + 1.$$

This method of coding is also known as Shannon coding and predates Huffman coding.

PROBLEM 4.

(a) $H(X) = \frac{2}{3}\log\frac{3}{2} + \frac{1}{3}\log 3 = 0.918$ bits $= H(Y)$.

(b) $H(X|Y) = \frac{1}{3}H(X|Y=0) + \frac{2}{3}H(X|Y=1) = 0.667$ bits $= H(Y|X)$.

(c) $H(X,Y) = 3 \times \frac{1}{3}\log 3 = 1.585$ bits.

(d) $H(Y) - H(Y|X) = 0.251$ bits.

(e) $I(X;Y) = H(Y) - H(Y|X) = 0.251$ bits.

PROBLEM 5.

(a) Let X be the number of tosses until the first head appears in a sequences of independent coin tosses, suppose the coin lands heads with probability p, and tails with probability q. Then $X = n$ if and only if the first $n-1$ tosses are tails and the last one is a head. Thus $\Pr(X = n) = pq^{n-1}$, $n = 1, 2, \ldots$. Then

$$\begin{aligned}
H(X) &= -\sum_{n=1}^{\infty} pq^{n-1}\log(pq^{n-1}) \\
&= -\left[\sum_{n=0}^{\infty} pq^n \log p + \sum_{n=0}^{\infty} npq^n \log q\right] \\
&= \frac{-p\log p}{1-q} - \frac{pq\log q}{p^2} \\
&= \frac{-p\log p - q\log q}{p} \\
&= H(p)/p \text{ bits.}
\end{aligned}$$

If $p = 1/2$, then $H(X) = 2$ bits.

(b) One possible questioning strategy is to ask the questions 'Is $X = 1$?', 'Is $X = 2$?', 'Is $X = 3$?', ..., stopping whenever a 'yes' answer is given. The number of questions asked when $X = n$ is exactly n, and thus the expected number of questions asked is $\sum_{n=1}^{\infty} n(1/2^n) = 2$.

Since this equals $H(X)$ this strategy cannot be improved upon.

Problem 6.

$$\begin{aligned} H(X) &= -\sum_{k=1}^{M} P_X(a_k) \log P_X(a_k) \\ &= -\sum_{k=1}^{M-1} (1-\alpha) P_Y(a_k) \log[(1-\alpha) P_Y(a_k)] - \alpha \log \alpha \\ &= (1-\alpha) H(Y) - (1-\alpha) \log(1-\alpha) - \alpha \log \alpha \end{aligned}$$

Since Y is a random variable that takes $M-1$ values $H(Y) \leq \log(M-1)$ with equality if and only if Y takes each of its possible values with equal probability.

Problem 7.

(a) Using the chain rule for mutual information,

$$I(X, Y; Z) = I(X; Z) + I(Y; Z \mid X) \geq I(X; Z),$$

with equality iff $I(Y; Z \mid X) = 0$, that is, when Y and Z are conditionally independent given X.

(b) Using the chain rule for conditional entropy,

$$H(X, Y \mid Z) = H(X \mid Z) + H(Y \mid X, Z) \geq H(X \mid Z),$$

with equality iff $H(Y \mid X, Z) = 0$, that is, when Y is a function of X and Z.

(c) Using the chain rule for mutual information,

$$I(X; Z \mid Y) + I(Z; Y) = I(X, Y; Z) = I(Z; Y \mid X) + I(X; Z)\,,$$

and therefore

$$I(X; Z \mid Y) = I(Z; Y \mid X) - I(Z; Y) + I(X; Z)\,.$$

We see that this inequality is actually an equality in all cases.

(d) Using first the chain rule for entropy and then the definition of conditional mutual information,

$$\begin{aligned} H(X, Y, Z) - H(X, Y) &= H(Z \mid X, Y) = H(Z \mid X) - I(Y; Z \mid X) \\ &\leq H(Z \mid X) = H(X, Z) - H(X)\,, \end{aligned}$$

with equality iff $I(Y; Z \mid X) = 0$, that is, when Y and Z are conditionally independent given X.