ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 9 — Information Theory and Coding

Solutions to homework 4 — October 22, 2013

Problem 1.

(a) Consider the sequence $a_i = 2^i - 1$, $i = 0, 1, 2, \ldots$ This is a strictly increasing sequence with $a_0 = 0, a_1 = 1, a_2 = 3, a_3 = 7, \ldots$ consequently any $M > 0$ will fall between two unique consecutive terms of this sequence, $a_k \leq M < a_{k+1}$, i.e., $M = a_k + r$, with $0 \leq r < a_{k+1} - a_k = 2^k$. This concludes the existence part.

For the uniqueness part, suppose that there exists another pair of integers (k', r') satisfying $M = 2^{k'} - 1 + r'$ and $0 \leq r' < 2^{k'}$. This means that $2^{k'} - 1 \leq M < 2^{k'+1} - 1$. Therefore, $k' = k$, from which we can easily deduce that $r = r'$.

(b) Consider a non-singular code that maximizes the Kraft sum $K = \sum_i 2^{-l_i}$. Let L be the length of the longest codeword in such a code. If the tree representing the code contains a node at a level $l < L$ which is not occupied by any codeword, then by deleting any codeword of length L and replacing it by the unoccupied node at level $l < L$ we obtain a new code with a higher Kraft sum which is a contradiction. Therefore, all the levels that are below the level L are completely occupied. This means that the number of codewords of length at most $L - 1$ is exactly $2^L - 1$ and the number of codewords of length L is $N_L \leq 2^L$. We conclude that $M = 2^L - 1 + N_L$ and $1 \leq N_L \leq 2^L$. We have two cases:

(i) $N_L = 2^L$, which means that $M = 2^{L+1} - 1$ so that $k = L + 1 = \log_2(M + 1) = \lceil \log_2(M + 1) \rceil$ and $r = 0$. In this case we have $K = \sum_i 2^{-l_i} = L + 1 = k = k + r2^{k-1} = \lceil \log_2(M + 1) \rceil$.

(ii) $N_L < 2^L$, which means that $k = L$ and $r = N_L$ (because of (a)). In this case, we have $K = \sum_i 2^{-l_i} = k + r2^{k-1} \leq k + 1 = \lceil \log_2(M + 1) \rceil$.

In both cases, we have $\sum_i 2^{-l_i} = k + r2^{k-1} \leq \lceil \log_2(M + 1) \rceil$. And since the non-singular code was chosen to maximize the Kraft sum, we conclude that any non-singular code satisfies $\sum_i 2^{-l_i} \leq k + r2^{k-1} \leq \lceil \log_2(M + 1) \rceil$.

(c) Define $K = \sum_i 2^{-l_i}$ and for each symbol a_i define $q(a_i) = \frac{2^{-l_i}}{K}$. It is clear that q is a probability distribution over the alphabet $\{a_1, ..., a_M\}$. Let p be the probability distribution of the random variable U. By the positivity of the kullback-leibler divergence, we have:

$$D(p||q) = \sum_{i=1}^{M} p(a_i) \log_2 \frac{p(a_i)}{q(a_i)} \geq 0,$$

from which we conclude that

$$\sum_{i=1}^{M} p(a_i) \log_2 p(a_i) - \sum_{i=1}^{M} p(a_i) \log_2 \frac{2^{-l_i}}{K} \geq 0.$$

By rewriting the last inequality we get $-H(U) + \bar{l} + \log_2(K) \geq 0$, and by applying the inequalities of part (b), we conclude:

$$\bar{l} \geq H(U) - \log_2(K) \geq H(U) - \log_2(k + r2^{-k}) \geq H(U) - \log_2 \lceil \log_2(M + 1) \rceil.$$

PROBLEM 2.

(a) Let $l_i = \lceil \log_2 \frac{2}{P_1(a_i)+P_2(a_i)} \rceil$, and let us compute the Kraft sum associated to $(l_i)_i$:

$$\sum_{i=1}^{M} 2^{-l_i} \leq \sum_{i=1}^{M} 2^{-\log_2 \frac{2}{P_1(a_i)+P_2(a_i)}} = \sum_{i=1}^{M} \frac{P_1(a_i)+P_2(a_i)}{2} = 1.$$

Since the Kraft sum is at most 1, there exists a prefix-free code where the length of the codeword associated to a_i is l_i.

(b) Since the code constructed in (a) is prefix free, it must be the case that $\bar{l} \geq H(U)$. In order to prove the upper bound, let P^* be the true distribution (which is either P_1 or P_2). It is easy to see that $P^*(a_i) \leq P_1(a_i) + P_2(a_i)$ for all $1 \leq i \leq M$. We have:

$$\begin{aligned}
\bar{l} &= \sum_{i=1}^{M} P^*(a_i).l_i = \sum_{i=1}^{M} P^*(a_i).\Big\lceil \log_2 \frac{2}{P_1(a_i)+P_2(a_i)} \Big\rceil \\
&< \sum_{i=1}^{M} P^*(a_i).\Big(1 + \log_2 \frac{2}{P_1(a_i)+P_2(a_i)}\Big) = \sum_{i=1}^{M} P^*(a_i).\Big(2 + \log_2 \frac{1}{P_1(a_i)+P_2(a_i)}\Big) \\
&= 2 + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{P_1(a_i)+P_2(a_i)} \overset{(*)}{\leq} 2 + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{P^*(a_i)} = 2 + H(U),
\end{aligned}$$

where the inequality $(*)$ uses the fact that $P^*(a_i) \leq P_1(a_i) + P_2(a_i)$ for all $1 \leq i \leq M$.

(c) Now let $l_i = \lceil \log_2 \frac{k}{P_1(a_i)+\ldots+P_k(a_i)} \rceil$, and let us compute the Kraft sum associated to $(l_i)_i$:

$$\sum_{i=1}^{M} 2^{-l_i} \leq \sum_{i=1}^{M} 2^{-\log_2 \frac{k}{P_1(a_i)+\ldots+P_k(a_i)}} = \sum_{i=1}^{M} \frac{P_1(a_i)+\ldots+P_k(a_i)}{k} = 1.$$

Since the Kraft sum is at most 1, there exists a prefix-free code where the length of the codeword associated to a_i is l_i. Since the code is prefix free, it must be the case that $\bar{l} \geq H(U)$. In order to prove the upper bound, let P^* be the true distribution (which is either P_1 or ... or P_k). It is easy to see that $P^*(a_i) \leq P_1(a_i) + \ldots + P_k(a_i)$ for all $1 \leq i \leq M$. We have:

$$\begin{aligned}
\bar{l} &= \sum_{i=1}^{M} P^*(a_i).l_i = \sum_{i=1}^{M} P^*(a_i).\Big\lceil \log_2 \frac{k}{P_1(a_i)+\ldots+P_k(a_i)} \Big\rceil \\
&< \sum_{i=1}^{M} P^*(a_i).\Big(1 + \log_2 \frac{k}{P_1(a_i)+\ldots+P_k(a_i)}\Big) \\
&= \sum_{i=1}^{M} P^*(a_i).\Big(1 + \log_2 k + \log_2 \frac{1}{P_1(a_i)+\ldots+P_k(a_i)}\Big) \\
&= 1 + \log_2 k + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{P_1(a_i)+\ldots+P_k(a_i)} \\
&\overset{(*)}{\leq} 1 + \log_2 k + \sum_{i=1}^{M} P^*(a_i).\log_2 \frac{1}{P^*(a_i)} = 1 + \log_2 k + H(U),
\end{aligned}$$

where the inequality $(*)$ uses the fact that $P^*(a_i) \leq P_1(a_i) + \ldots + P_k(a_i)$ for all $1 \leq i \leq M$.

Problem 3.

(a) Consider a maximally branched Huffman code, and for each $1 \le l \le l_{\max}$, let N_l be the number of codewords of length l. Since the Huffman code is maximally branched, we have $N_l \ge 1$ for $1 \le l < l_{\max}$, and clearly we have $N_l \ge 2$ for $l = l_{\max}$ since any Huffman code contains at least two longest codewords. The Kraft-sum of this code is equal to:

$$\sum_{l=1}^{l_{\max}} N_l 2^{-l} \ge \Big(\sum_{l=1}^{l_{\max}-1} 2^{-l} \Big) + 2.2^{-l_{\max}} = 2^{-1}\frac{1-2^{-l_{\max}+1}}{2^{-1}} + 2^{-l_{\max}+1} = 1,$$

where the equality holds if and only if we have $N_l = 1$ for $1 \le l < l_{\max}$ and $N_{l_{\max}} = 2$. Now since any Huffman code is a prefix-free code, the Kraft-sum must be at most 1. We conclude that the Kraft-sum is equal to 1, which implies that $N_l = 1$ for $1 \le l < l_{\max}$ and $N_{l_{\max}} = 2$.

(b) We will prove by induction on $M \ge 3$ the following statement: If we have $P(a_i) \ge \sum_{j=1}^{i-2} P(a_j)$ for every $3 \le i \le M$, there exists a maximally branched Huffman code in which the codewords associated to a_1 and a_2 are the longest two codewords. The statement is trivial for $M = 3$. Now suppose that the statement is true up to alphabets of length $M-1$, and suppose that we have an alphabet of length $M > 3$ such that $P(a_i) \ge \sum_{j=1}^{i-2} P(a_j)$ for every $3 \le i \le M$. Now consider the alphabet $\{a'_1, \dots, a'_{M-1}\}$ such that $a'_i = a_{i+1}$ for $2 \le i \le M-1$, and define the probability distribution P' on this alphabet by $P'(a'_1) = P(a_1) + P(a_2)$ and $P'(a'_i) = P(a'_i) = P(a_{i+1})$ for every $2 \le i \le M-1$. It is easy to show that we have $P'(a'_i) \ge \sum_{j=1}^{i-2} P(a'_j)'$ for every $3 \le i \le M-1$. By the induction hypothesis, there exists a maximally branched Huffman code for the new alphabet in which the codewords associated to a'_1 and a'_2 are the longest two words. By deleting the codeword associated to a'_1 and replacing it with its two descendants, and associating the new codewords to a_1 and a_2, we get a maximally branched Huffman code for the original alphabet $\{a_1, \dots, a_M\}$ in which the codewords associated to a_1 and a_2 are the longest two codewords.

(c) We will prove the statement by induction on $M \ge 3$. The statement is trivial for $M = 3$. Now suppose that it is true for alphabets of length up to $M-1$, and consider an alphabet of length M satisfying $P(a_i) > \sum_{j=1}^{i-2} P(a_j)$ for every $3 \le i \le M$. It is easy to see that a_1 and a_2 are the unique two symbols with smallest probability, and so every Huffman code must begin by combining a_1 and a_2. Now consider the alphabet $\{a'_1, \dots, a'_{M-1}\}$ such that $a'_i = a_{i+1}$ for $2 \le i \le M-1$, and define the probability distribution P' by $P'(a'_1) = P(a_1) + P(a_2)$ and $P'(a'_i) = P(a'_i) = P(a_{i+1})$ for every $2 \le i \le M-1$. It is easy to show that we have $P'(a'_i) > \sum_{j=1}^{i-2} P(a'_j)'$ for every $3 \le i \le M-1$. Since every Huffman code for the new alphabet is maximally branched, every Huffman code for the initial alphabet $\{a_1, \dots, a_M\}$ is maximally branched as well.

(d) Let $P(a_i)=\frac{\varphi^i}{\sum_{j=1}^M \varphi^j}=\frac{\varphi^{i-1}(\varphi-1)}{\varphi^M-1}$. It is easy to see that $P(a_1)\leq\ldots\leq P(a_M)$. We will prove by induction on $3\leq i\leq M$ that we have $\displaystyle\sum_{j=1}^{i-2}P(a_j)=P(a_i)$. The statement is trivial for $i=3$ since $\varphi^2=\varphi+1$. Now let $4\leq i\leq M$ and suppose that we have $\displaystyle\sum_{j=1}^{i-3}P(a_j)=P(a_{i-1})$, then:

$$\sum_{j=1}^{i-2}P(a_j)=P(a_{i-2})+\sum_{j=1}^{i-3}P(a_j)=P(a_{i-2})+P(a_{i-1})=\frac{(\varphi^{i-3}+\varphi^{i-2})(\varphi-1)}{\varphi^M-1}$$
$$=\frac{\varphi^{i-3}(1+\varphi)(\varphi-1)}{\varphi^M-1}=\frac{\varphi^{i-3}(\varphi^2)(\varphi-1)}{\varphi^M-1}=\frac{\varphi^{i-1}(\varphi-1)}{\varphi^M-1}=P(a_i).$$

By applying (b), we get the result.
Note that the Huffman code for this distribution has $l_1=M-1$, where as $\log_2\frac{1}{p_1}=M\log_2\phi-\text{const}\approx(0.695)M-\text{const}$. We see that l_1 and $\log_2\frac{1}{p_1}$ can be very different. Therefore, it is not true that l_i is close to $\log_2\frac{1}{p_i}$ for Huffman codes.

Problem 4.

(a) We prove the identity by induction on $n\geq 1$. For $n=1$, the identity is trivial. Let $n>1$ and suppose that the identity is true up to $n-1$. We have:

$$I(Y_1^{n-1};X_n)=I(Y_1^{n-2},Y_{n-1};X_n)\overset{(*)}{=}I(Y_1^{n-2};X_n)+I(X_n;Y_{n-1}|Y_1^{n-2})$$
$$\overset{(**)}{=}\Big(\sum_{i=1}^{n-2}I(X_n;Y_i|Y_1^{i-1})\Big)+I(X_n;Y_{n-1}|Y_1^{n-2})=\sum_{i=1}^{n-1}I(X_n;Y_i|Y_1^{i-1}).$$

The identity $(*)$ is by the chain rule for mutual information, and the identity (**) is by the induction hypothesis.

(b) For every $0\leq i\leq n$, define $a_i=I(X_{i+1}^n;Y_1^i)$, and for every $1\leq i\leq n$, define $b_i=I(X_{i+1}^n;Y_1^{i-1})$. It is easy to see that $a_0=a_n=0$. We have:

$$\sum_{i=1}^{n}I(X_{i+1}^n;Y_i|Y_1^{i-1})\overset{(*)}{=}\sum_{i=1}^{n}\Big(I(X_{i+1}^n;Y_1^i)-I(X_{i+1}^n;Y_1^{i-1})\Big)=\Big(\sum_{i=1}^{n}a_i\Big)-\Big(\sum_{i=1}^{n}b_i\Big)$$
$$\overset{(**)}{=}\Big(\sum_{i=0}^{n-1}a_i\Big)-\Big(\sum_{i=1}^{n}b_i\Big)=\Big(\sum_{i=1}^{n}a_{i-1}\Big)-\Big(\sum_{i=1}^{n}b_i\Big)=\sum_{i=1}^{n}\Big(a_{i-1}-b_i\Big)$$
$$=\sum_{i=1}^{n}\Big(I(X_i^n;Y_1^{i-1})-I(X_{i+1}^n;Y_1^{i-1})\Big)\overset{(***)}{=}\sum_{i=1}^{n}I(Y_1^{i-1};X_i|X_{i+1}^n).$$

The identities $(*)$ and $(***)$ are by the chain rule for mutual information. The identity $(**)$ follows from the fact that $a_0=a_n=0$, which implies that $\displaystyle\sum_{i=1}^{n}a_i=\sum_{i=0}^{n-1}a_i$.