Information Theory I

Fall Semester 2017

Prof. Dr. A. Lapidoth

Signal and Information Processing Laboratory
Institut für Signal- und Informationsverarbeitung

Model Answers to Exercise 4 of October 11, 2017

http://www.isi.ee.ethz.ch/teaching/courses/it1.html

Problem 1 — *Slackness in Kraft's Inequality*

a) The codewords of a prefix-free code can be assigned to internal nodes or leaves of a binary tree of depth $l_{\max} = \max\{l_1, l_2, \ldots, l_m\}$ such that a codeword of length l_i is assigned to a node at depth l_i and all the children of this node cannot represent any other codeword. We will describe the latter fact by saying that the $2^{l_{\max}-l_i}$ descending leaves are shadowed, which is depicted in Fig. 1 (i) using dotted lines and gray colored leaves. (In case the node representing the codeword is a leaf, we will also say that the leaf is shadowed.)

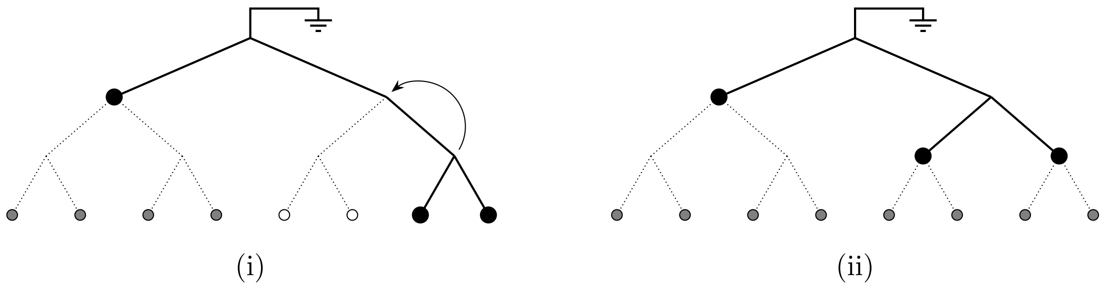

Figure 1: A tree representing a prefix-free code satisfying Kraft's inequality with strict inequality is depicted in (i), and a tree corresponding to a deterministically better prefix-free code is shown in (ii). Nodes corresponding to codewords are black, shadowed leaves (unless they correspond to codewords) are gray, and unshadowed leaves are white.

The tree consists of $2^{l_{\max}}$ leaves, and the total number of shadowed leaves is $\sum_{i=1}^{m} 2^{l_{\max}-l_i}$. We have

$$\sum_{i=1}^{m} 2^{l_{\max}-l_i} = 2^{l_{\max}} \sum_{i=1}^{m} 2^{-l_i} \overset{(*)}{<} 2^{l_{\max}} \cdot 1 = 2^{l_{\max}},$$

where (*) holds because Kraft's inequality holds with strict inequality. Hence, there must exist at least one leaf which is not shadowed by any codeword, i.e., for which there are no codewords assigned on the path from the leaf to the root. Consequently, we can improve our code in the following way, which is illustrated in Fig. 1:

- Choose one of the leaves that is not shadowed and follow its path up to the root until you encounter a node whose subtree contains a codeword;
- exactly one child of that node has a subtree which contains codewords, so remove the node and replace it with that child. Thus, the codeword length is reduced for at least one source symbol; and no source symbol gets a longer codeword length.

(Alternatively, the problem can be solved as follows: Let again $l_{\max} \triangleq \max\{l_1, l_2, \ldots, l_m\}$, and assume without loss of generality that $l_m = l_{\max}$. Because Kraft's inequality holds with strict inequality, we have

$$\sum_{i=1}^{m} 2^{l_{\max}-l_i} = 2^{l_{\max}} \sum_{i=1}^{m} 2^{-l_i} < 2^{l_{\max}} \cdot 1 = 2^{l_{\max}}.$$

Observe that both sides of the inequality are integers, so

$$\sum_{i=1}^{m} 2^{l_{\max}-l_i} \leq 2^{l_{\max}} - 1$$

must hold, which is equivalent to

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1 - 2^{-l_{\max}}. \tag{1}$$

Reduce the codeword length of symbol m by one, i.e., define new codewords lengths $\tilde{l}_1 \triangleq l_1$, $\tilde{l}_2 \triangleq l_2, \ldots, \tilde{l}_{m-1} \triangleq l_{m-1}, \tilde{l}_m \triangleq l_m - 1$. The codeword lengths $\tilde{l}_1, \tilde{l}_2, \ldots, \tilde{l}_m$ still satisfy Kraft's inequality:

$$\begin{aligned} \sum_{i=1}^{m} 2^{-\tilde{l}_i} &= \sum_{i=1}^{m-1} 2^{-\tilde{l}_i} + 2^{-\tilde{l}_m} \\ &= \sum_{i=1}^{m-1} 2^{-l_i} + 2^{-(l_m-1)} \\ &= \sum_{i=1}^{m-1} 2^{-l_i} + 2^{-l_m} + 2^{-l_m} \\ &\stackrel{\text{(i)}}{=} \sum_{i=1}^{m} 2^{-l_i} + 2^{-l_{\max}} \\ &\stackrel{\text{(ii)}}{\leq} 1, \end{aligned}$$

where (i) holds because $l_m = l_{\max}$; and (ii) follows from (1). Because Kraft's inequality is satisfied, a prefix-free code with codeword lengths $\tilde{l}_1, \tilde{l}_2, \ldots, \tilde{l}_m$ exists, and such a code is deterministically better than the original code.)

b) In general, the conclusion does not hold for $\mathsf{D} > 2$. A simple example is the following: Let $\mathsf{D} > 2$, let $m = 2$, and let $l_1 = l_2 = 1$. Then, Kraft's inequality holds with strict inequality:

$$\sum_{i=1}^{2} \mathsf{D}^{-l_i} = 2 \cdot \frac{1}{\mathsf{D}} < 1,$$

but it is not possible to find a prefix-free code that is deterministically better, since $l_1 \geq 1$ and $l_2 \geq 1$ must hold for every prefix-free code.

Problem 2 ***Shannon Code***

a) We have:

$$\begin{aligned} l_1 &= 1 \qquad \mathsf{F}_1 = 0 &&= 0.\underline{0}000_2 \\ l_2 &= 2 \qquad \mathsf{F}_2 = 0.5 &&= 0.\underline{10}00_2 \\ l_3 &= 3 \qquad \mathsf{F}_3 = 0.75 &&= 0.\underline{110}0_2 \\ l_4 &= 3 \qquad \mathsf{F}_4 = 0.875 &&= 0.\underline{111}0_2 \end{aligned}$$

The codewords are 0, 10, 110, and 111.

b) We prove by contradiction that the code generated by this process is prefix-free. Denote the codeword of the i-th symbol by c_i. Assume that the code is not prefix-free, i.e., that there exist integers i and j with $1 \leq i < j \leq m$ such that c_i is a prefix of c_j or that c_j is a prefix of c_i. We have $p_i \geq p_j$ because the probabilities are ordered and because $i < j$. Consequently, $l_i = \lceil \log_2 \frac{1}{p_i} \rceil \leq \lceil \log_2 \frac{1}{p_j} \rceil = l_j$ holds. Observe that if c_j is a prefix of c_i, then $c_i = c_j$ holds; in this case, c_i is also a prefix of c_j. It thus suffices to check whether c_i is a prefix of c_j.

In order for c_i to be a prefix of c_j, the first l_i bits of F_i and F_j must agree. Because $i < j$,

$$\mathsf{F}_j - \mathsf{F}_i = \sum_{k=1}^{j-1} p_k - \sum_{k=1}^{i-1} p_k = \sum_{k=i}^{j-1} p_k \geq p_i. \tag{2}$$

Because $l_i = \lceil \log_2 \frac{1}{p_i} \rceil \geq \log_2 \frac{1}{p_i}$, we have

$$p_i \geq 2^{-l_i}. \tag{3}$$

Combining (2) and (3) leads to the desired contradiction: since $\mathsf{F}_j - \mathsf{F}_i \geq 2^{-l_i}$, it is not possible that the first l_i bits of F_i and F_j agree. We thus conclude that the code is always prefix-free.

The expected length of the code can be bounded as follows:

$$\begin{aligned} \mathsf{L} &= \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} p_i \left\lceil \log_2 \frac{1}{p_i} \right\rceil < \sum_{i=1}^{m} p_i \left[\log_2 \frac{1}{p_i} + 1 \right] = H(X) + 1, \\ \mathsf{L} &= \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} p_i \left\lceil \log_2 \frac{1}{p_i} \right\rceil \geq \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} = H(X). \end{aligned}$$

Problem 3 ***Optimal Code Lengths that Require One Bit above Entropy***

For $\delta \in (0,1)$, let X be Bernoulli(δ). The entropy of X is $H_\mathrm{b}(\delta)$. The code $0 \mapsto 0, 1 \mapsto 1$ is a uniquely decodable code with expected length $\mathsf{L} = 1$. This code is optimal because every codeword of a uniquely decodable code must have positive length.
Fix $\epsilon > 0$. Because the binary entropy function is continuous and because $H_\mathrm{b}(0) = 0$, there must exist a $\delta \in (0,1)$ with $H_\mathrm{b}(\delta) < \epsilon$. For such a δ, the above choice of X leads to

$$H(X) + 1 - \epsilon = H_\mathrm{b}(\delta) + 1 - \epsilon < 1 = \mathsf{L}^*.$$

Consequently, for every $\epsilon > 0$, there exists a chance variable X for which $\mathsf{L}^* > H(X) + 1 - \epsilon$.