Class 17

Example for Shannon-Fano code:

$X \in \mathcal{X} = \{x_1, x_2, x_3, x_4\}$

$$p_X(x_i) = \begin{cases} 1/4 & i=1 \\ 1/2 & i=2 \\ 1/9 & i=3 \\ 5/36 & i=4 \end{cases}$$

Lengths satisfying the Kraft for the S-F code

$$l_i = \left\lceil \log_2 \frac{1}{p_X(x_i)} \right\rceil = \begin{cases} 2, & \text{if } i=1 \\ 1, & \text{if } i=2 \\ 4, & \text{if } i=3 \\ \left\lceil \log_2 36/5 \right\rceil = 3, & \text{if } i=4 \end{cases}$$

↑ avg length of S-F code.

$$\bar{L} = \sum_{i=1}^{4} p_i l_i = \frac{1}{4}\times 2 + \frac{1}{2}\times 1 + \frac{1}{9}\times 4 + \frac{5}{36}\times 3$$

$$= 57/36.$$

Ex: Check that these lengths satisfy Kraft inequality $\sum 2^{-l_i} \le 1$

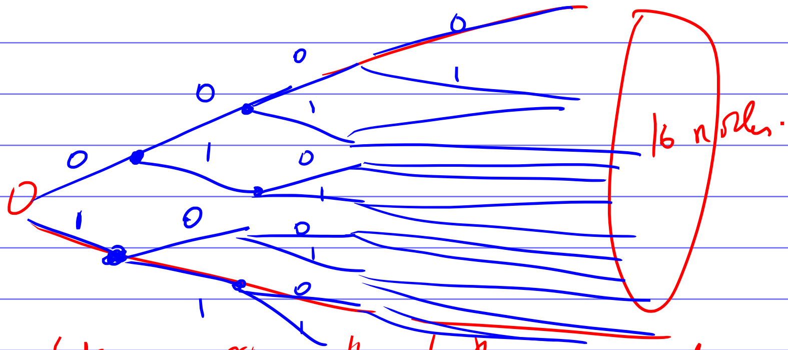
& verify that (a) $\bar{L} \ge H(X)$ (b) $< H(X)+1$.

Find $H(X) = \sum_{i=1}^{4} p_i \log \frac{1}{p_i}$

Ex:

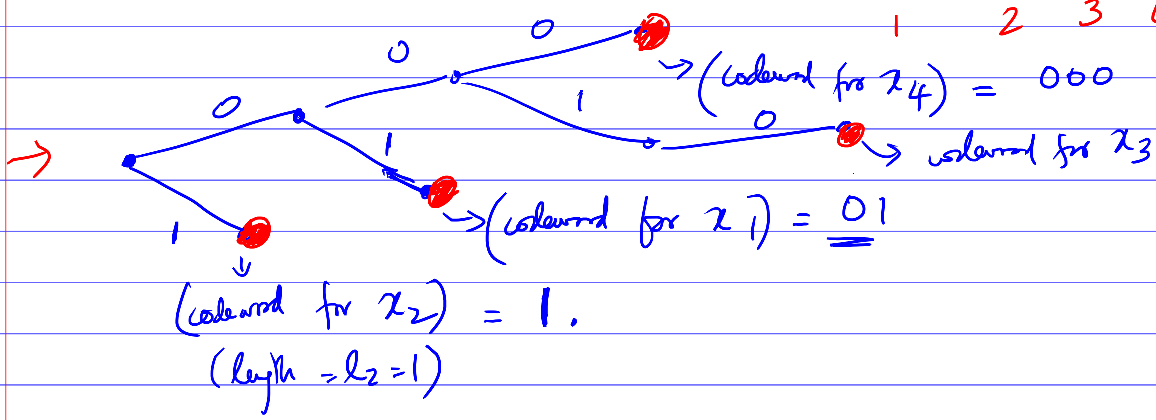
Getting the S-F code: (algorithm)

→ Consider a binary tree upto depth 4.



→ Let us arrange the lengths in ascending $l_2 \le l_1 \le l_4 \le l_3$

$l_2 = 1,\ l_1 = 2,\ l_4 = 3,\ l_3 = 4$



$x_1 \to \{01,$

$x_2 \to 1,$

$x_3 \to 0010,$

$x_4 \to 000\} \longrightarrow$ S-F code for X. [This code is P-F by the algorithm]

Note: The SF code is not optimal (in general) [here 'optimal' means having the smallest avg length $\bar{L}$ among all possible codes for X which are P-F]

→ To prove this we will give an optimal code construction & show that the SF code has a larger avg length.

→ Optimal code for Source Coding (Fixed to Variable lengths) is called Huffman coding. [invented by Huffman].

Some intuitive lemmas about an optimal (PF) code for RV X:

Assume that $X \in \mathcal{X} = \{x_1, \dots, x_K\}$.

$P_X(x_i) = p_i$, $i = 1 \dots K$.

WLOG: $p_1 \geq \dots \geq p_K$ → (A)

Lemma 1: Consider that $l_1, \dots, l_K$ are the lengths of the codewords associated to the msgs x_i : $i = 1, \dots K$ respectively in any optimal code for X.

Then $l_1 \leq l_2 \leq \dots \leq l_{K-1} \leq l_K$ → (B)

Proof: Assumed statement

Suppose $\mathcal{C}$ is a (optimal) code in which $\exists$ some distinct $i, j \in \{1, \dots, K\}$ such that $p_i > p_j$ but $l_i > l_j$. [Assumption of the contrary to statement we want to prove (that is, (B))]

We will show that there is another code $\mathcal{C}'$ which has smaller avg length $L_{\mathcal{C}'}$ than $L_{\mathcal{C}}$.

This means $\mathcal{C}$ is NOT optimal ⇒ Contradiction with the assumed statement

Consider the code $\mathcal{C}'$ in which the codewords for x_i & x_j are swapped bw those in $\mathcal{C}$.
Keep all other mappings & codewords as the same as in $\mathcal{C}$

$\underline{C}_{\mathcal{C}'}(x_j) = \underline{C}_{\mathcal{C}}(x_i)$ & $\underline{C}_{\mathcal{C}'}(x_i) = \underline{C}_{\mathcal{C}}(x_j)$

↑ codeword for x_j in $\mathcal{C}'$; ↓ codeword for x_j in $\mathcal{C}$.

Now in $\mathcal{C}'$ $l_{i,\mathcal{C}'} \triangleq$ length of codeword associated to x_i in $\mathcal{C}'$

$= l_j$

$l_{j,\mathcal{C}'} = l_i$.

So C' is barely the same codewords as C & hence is also prefix-free.

$$\bar{L}_{C'} = \sum_{k \in \{1,\cdots,K\} \setminus \{i,j\}} p_k l_k + p_i l_j + p_j l_i$$

$$\bar{L}_{C} = \sum_{k \in \{1,\cdots,K\}} p_k l_k$$

$$\bar{L}_{C'} - \bar{L}_{C} = p_i (l_j - l_i) + p_j (l_i - l_j)$$

$$= \underline{(l_j - l_i)} \; \underline{(p_i - p_j)} \rightarrow \text{positive}$$

We have $p_i > p_j$, but $l_i > l_j$ → hence this is negative

$$\Rightarrow \bar{L}_{C'} - \bar{L}_{C} < 0 \Rightarrow \bar{L}_{C'} < \bar{L}_{C}$$

$\Rightarrow$ C is not optimal.

Thus there is a contradiction

$\Rightarrow$ 'Contrary to (B)' cannot happen for a optimal code

$\Rightarrow$ (B) has to happen for an optimal code.

PF

Lemma 2: Consider the tree representation of a optimal code.

In such a tree, it should be true that

either each node is a codeword

(or) it has atleast 2 successors which are codewords

"There are no unused leaves in the tree of an optimal code"

Example: