EL-GY 6063: Information Theory
Lecture 4

March 1, 2019

1 Lossless Source Coding

References: Chapter 5 of textbook.

Definition 1. *Let D^* be the set of all finite sequences of binary variables. That is $D^* = \{x^n | n \in \mathbb{N}, x_i \in \{0,1\}\}$. Then, a source code for the random vector X^n is the mapping $C : \mathcal{X}^n \to D^*$. $C(x^n)$ is called the codeword representing x^n. $l(x^n)$ is the length of the sequence $C(x^n)$.*

Definition 2. *The expected length of the Code C for the random vector X^n is defined as $\bar{l} = \sum_{x^n} P_{X^n}(x^n) l(x^n)$.*

Definition 3. *A code is called non-singular if every sequence x^n is mapped to a different codeword. That is $x^n \neq x^{'n} \Rightarrow C(x^n) \neq C(x^{'n})$.*

Definition 4. *The extension C^* of a code is concatenation of the code with itself. That is $\forall k : C^*(x_1^n, x_2^n, \cdots, x_k^n) = C(x_1^n)C(x_1^n)\cdots C(x_k^n)$.*

Definition 5. *A code is called uniquely decodable if its extension is non-singular.*

Definition 6. *A codeword $C(x^n)$ is called a prefix of another codeword $C(x^{'n})$ if it is the first part of $C(x^{'n})$.*

Example: Assume that $C(x^n) = 01$ and $C(x^{'n}) = 011$. Then, it is a prefix.

Definition 7. *A code is called a prefix-free code (prefix code or instantaneous code) if no codeword is a prefix of another codeword.*

Theorem 1. *(Kraft Inequality) For any prefix code for a source X with alphabet size m, with codeword lengths $l_1, l_2, \cdots, l_m$, the following inequality is satisfied:*

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

Conversely, for any given vector of positive integers $(l_1, l_2, \cdots, l_m)$ satisfying the Kraft inequality, there exists a prefix code with these codeword lengths.

proof. The proof is based on a structure called the binary code tree. Assume that we draw the tree with depth equal $l_{max} = max l_i$. For a prefix code no code is a child or grandchild of another code in the binary tree. Note that a codeword of length l_i has $2^{l_{max}-l_i}$ leaves connected to it. The total number of leaves is equal to $2^{l_{\max}}$. So, we have to have $\sum_{i=1}^{n} 2^{l_{max}-l_i} \leq 2^{l_{max}}$. This gives the kraft inequality.

The converse proof also follows by looking at the binary tree. We construct the code in the following way. First, we go l_1 steps down the first branch. We pick the child as the first codeword. Then, we go l_2 steps down next branch, name child the second codeword. Continue until all codewords assigned. Each step we are eliminating $2^{l_{max}-l_i}$ leaves. We can complete the process as long as leaves remain. So, we can construct if the Kraft inequality is satisfied.

Example: Assume that for a source X, the distribution is such that $-\log P_X(x)$ are all natural numbers. Then, a code with $l_x = -\log P_X(x), x \in \mathcal{X}$ exists.

proof. $\sum_{x\in|\mathcal{X}|} 2^{-l_i} = \sum_x 2^{\log P_X(x))} = \sum_x P_X(x) = 1$.

Note that $\bar{l} = \mathbb{E}(l_x) = \mathbb{E}(-\log P_X(x)) = H(X)$.

So, if $-\log P_X(x)$ are all natural numbers then a code with expected length equal to the entropy exists.

Theorem 2. *(McMillan) For any uniquely decodable code for a source X with alphabet size m, with codeword lengths $l_1, l_2, \cdots, l_m$, the following inequality is satisfied:*

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

Conversely, for any given vector of positive integers $(l_1, l_2, \cdots, l_m)$ satisfying the above inequality, there exists a prefix code with these codeword lengths.

proof. The converse follows from the statement of the previous theorem. The result is given in book Th.5..5.1.

A code for the source X is called optimal if it has the minimum possible expected length among all codes for that source. Note that the optimal code may not be unique. The following result relates the expected length of the optimal code for source X to its entropy.

Theorem 3. *Let $l_1^*, l_2^*, \cdots, l_m^*$ be the codeword length for the optimal code C^* for the source X with alphabet size m. Let L^* be the expected length of the code ($L^* = \sum_{i=1}^m P_X(i) l_i^*$). Then,*

$$H(X) \leq L^* < H(X) + 1.$$

proof. First, we prove the left hand side. The proof follows from the Gibbs and Kraft inequalities. Let $Q_X(i) = \frac{2^{-l_i^*}}{\sum_{j=1}^m 2^{-l_i^*}}$ (as an exercise prove that this is a valid probability distribution.). Then, from the Gibbs inequality we have:

$$\sum_{i=1}^{m} P_X(i) \log \frac{P_X(i)}{Q_X(i)} \geq 0 \Rightarrow \sum_{i=1}^{m} P_X(i)(\log P_X(i) - \log Q_X(i)) \geq 0 \Rightarrow -H(X) - \sum_{i=1}^{m} P_X(i) \log \frac{2^{-l_i^*}}{\sum_{j=1}^m 2^{-l_i^*}} \geq 0$$

$$\Rightarrow -\sum_{i=1}^{m} P_X(i) \log 2^{-l_i^*} + \sum_{i=1}^{m} P_X(i) \log \sum_{j=1}^{m} 2^{-l_i^*} \geq H(X)$$

$$\Rightarrow L^* + \log \sum_{j=1}^{m} 2^{-l_i^*} \geq H(X) \Rightarrow L^* \geq H(X) - \log \sum_{j=1}^{m} 2^{-l_i^*}.$$

From the Kraft inequality, we know that $\sum_{j=1}^m 2^{-l_i^*} \leq 1$. So, $L^* \geq H(X)$ with equality if and only if $\sum_{j=1}^m 2^{-l_i^*} = 1$.

Note: Using Lagrange multipliers, one can show that $L^* = H(X)$ if and only if $-\log P_X(i), i = 1, 2, \cdots, m$ are all natural numbers. In this case $l_i^* = -\log P_X(i)$.

To prove the right hand side, we use Shannon-Fano Codes. Shannon-Fano codes have lengths $l_i = \lceil -\log P_X(i) \rceil, i = 1, 2, \cdots, m$, where $\lceil a \rceil$ is the smallest integer greater than a. It is straightforward to show that the lengths satisfy the Kraft inequality and hence there exist codes which are uniquely decodable with these lengths. Also, $\bar{l} = \sum_{i=1}^m P_X(i) l_i = \sum_{i=1}^m P_X(i) \lceil -\log P_X(i) \rceil \leq \sum_{i=1}^m P_X(i)(-\log P_X(i) + 1) \leq H(X) + 1$.

So, one can design codes whose average length comes to a 1 bit distance from entropy. The question is whether we can improve this distance. The improvement is achieved by using what is called block coding. In block coding several input symbols are codes simultaneously. (show figure)

Definition 8. *The expected length per input symbol of the Code C for the random vector X^n is defined as $\bar{l}_n = \frac{1}{n} \sum_{x^n} P_{X^n}(x^n) l(x^n)$. The code C is called a block code of length n for source X.*

Theorem 4. *For any block code of length n for the source X, the minimum expected length per symbol satisfies:*

$$H(X) \leq \bar{l}_n < H(X) + \frac{1}{n}.$$

proof. Assume that C_n^* is the optimal code for block coding a source sequence of length n with average length n. Then, from the previous theorem we have:

$$H(X^n) \leq l_n^* < H(X^n) + 1.$$

Note that $H(X^n) = nH(X)$ since X^n is an independent and identically distributed sequence. So,

$$nH(X) \leq l_n^* < nH(X) + 1 \Rightarrow H(X) \leq \bar{l}_n^* < H(X) + \frac{1}{n}.$$

Theorem 5. *Assume that X^n is a source which is not necessarily i.i.d and is distributed according to P_{X^n}, then the expected length per input symbol L_n satisfies the following:*

$$\underline{\frac{1}{n}H(X^n) \leq L_n^* \leq \frac{1}{n}H(X^n) + \frac{1}{n}}.$$

Example 1. *Assume that for the source X with distribution P_X on alphabet $\{1, 2, \cdots, m\}$, the approximation Q_X is available to the encoder. So, the encoder uses the Shannon-Fano code with lengths $l_i = \lceil -\log Q_X(i) \rceil$. We have:*

$$\bar{l} = \sum_{i=1}^{m} P_X(i) l_i = \sum_{i=1}^{m} P_X(i) \lceil -\log Q_X(i) \rceil \leq \sum_{i=1}^{m} -P_X(i) \log Q_X(i) + 1 = \sum_{i=1}^{m} P_X(i) \frac{1}{\log Q_X(i)} + 1$$
$$= \sum_{i=1}^{m} P_X(i) \frac{P_X(i)}{\log Q_X(i)} + H(X) + 1 = D(P_X || Q_X) + H(X) + 1.$$

Similarly, it can be shown that:

$$D(P_X || Q_X) + H(X) \leq \bar{l} \leq D(P_X || Q_X) + H(X) + 1.$$

So, the cost of approximating the distribution is $D(P_X || Q_X)$ bits per symbol.

2 Properties of Optimal Prefix Codes and Huffman Coding

A prefix code is called optimal if it has the minimum average length among all prefix codes. Note that there might be several prefix codes with the minimum length. In this case all of these codes are called optimal codes. One can prove the following properties for the optimal codes:

Theorem 6. *Let X be an information source with alphabet $\{x_1, x_2, \cdots, x_n\}$ and probabilities $(p_1, p_2, \cdots, p_n)$. Without loss of generality assume that $p_1 \geq p_2 \geq \cdots \geq p_n$. Assume C^* is an optimal prefix code for the source X with lengths $(l_1^*, l_2^*, \cdots, l_n^*)$, respectively (e.g. length of the codeword representing x_1 is l_1^*). Then, the following statements hold:*
1) The symbols with higher probabilities have shorter codewords: $p_i \leq p_j \Rightarrow l_i^* \geq l_j^*$.
2) The two least probability symbols have codewords of similar length: $l_{n-1}^* = l_n^*$
3) Among all codewords with the maximum length , at least two of them differ only in the last bit.

proof. 1) We prove the statement by contradiction. Assume that there exist i and j such that the statement does not hold. That is $p_i < p_j$ and $l_i^* < l_j^*$. Then, we construct a code C which has a lower average length than C^*. This contradicts the optimality of C^*. The code C assigns the same codewords to all symbols as C^* except for x_i and x_j. It assigns c_i^* to x_j and c_j^* to x_i. Let L^* be the average length of C^* and L be the average length of C. Then,

$$L^* - L = \sum_{k=1}^{n} p_k l_k^* - \left(\sum_{k=1, k \neq i,j}^{n} p_k l_k^* + p_i l_j^* + p_j l_i^* \right) = p_i l_i^* + p_j l_j^* - p_i l_j^* - p_j l_i^* = (p_i - p_j)(l_i^* - l_j^*) > 0.$$

This is a contradictions. So, the statement is proved.
2,3) This can be easily observed by considering the binary tree of the code. If the statement does not hold then the last bit in the longest codeword can be removed to construct a new code with a smaller average length. Once again, this contradicts optimality. So, the statement is correct.

Huffman Codes: We studied Huffman coding by solving Example 5.6.1 and 5.6.2 from the textbook.

Note: In order to improve the performance of Huffman codes, one could construct block Huffman codes as in the previous section.

Theorem 7. *Huffman codes are optimal prefix codes. That is for any other prefix code C' with average length L', we have $L' \geq L_H^n$ where L_H^n is the average length of Huffman codes for sources with alphabet size n.*

proof. We prove the claim by induction. Clearly, it is true for any source with alphabet size 2 (since there are only two possible codes for such a source and both have average length 1 the claim is automatically true for alphabet size 2). Assume that the claim is true for any source of alphabet size k. We prove that it is true for any source of alphabet size $k+1$. Assume that we are given the source X with alphabet $\{x_1, x_2, \cdots, x_{k+1}\}$ and probabilities $p_1, p_2, \cdots, p_{k+1}$ such that $p_1 \geq p_2 \geq \cdots p_{k+1}$. Let C^* be an optimal code for X. Since C^* is optimal, we know that $L^* \leq L_H^{k+1}$ so $\sum_{i=1}^{k+1} p_i l_i \leq \sum_{i=1}^{k+1} p_i l_i^H$. From the previous theorem we know that $l_k^* = l_{k+1}^*$ and that the codewords for x_k and x_{k+1} differ only in the last bit. Define a new random variable X' with alphabet $\{x_1', x_2', \cdots, x_k'\}$ with probabilities $p_1, p_2, \cdots, p_{k-1}, p_k + p_{k+1}$. We construct a code C' for X' using C^* as follows. We assign c_i to x_i for $i \leq k-1$ and we assign c_k' to x_k' where c_k' is equal to the sequence which results from removing the last bit from c_k (it could also be thought of as the codeword which results from removing the last bit from c_{k+1} since the two differ only in the last bit). It is straightforward to see that C' is a prefix code from the binary tree of the code. From the induction assumption that Huffman codes are optimal for all sources of alphabet size k, we know that $L' \geq L_H^k$. Note that:

$$L' = \sum_{i=1}^{k} p_i' l_i' = \sum_{i=1}^{k-1} p_i l_i + (p_k + p_{k+1})(l_k - 1) = \sum_{i=1}^{k+1} p_i l_i - p_k - p_{k-1} = L^* - p_k - p_{k-1}.$$

Similarly,

$$L_H^k = \sum_{i=1}^{k} p_i l_i^H = \sum_{i=1}^{k-1} p_i l_i^H + (p_k + p_{k+1})(l_k^H - 1) = \sum_{i=1}^{k+1} p_i l_i - p_k - p_{k-1} = L_H^{k+1} - p_k - p_{k-1}.$$

So, we have:

$$L' \geq L_H^k \Rightarrow L^* - p_k - p_{k-1} \geq L_H^{k+1} - p_k - p_{k-1} \Rightarrow L^* \geq L_H^{k+1}.$$