Last class, we looked at Huffman coding. To motivate today's topic, notice that when the probabilities $p(A_i)$ are all powers of two, for example, $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$. The codeword lengths turn out to be $\log \frac{1}{p(A_i)}$, i.e. 1,2,3,3, respectively. For more general probabilities, the the expression $\log \frac{1}{p(A_i)}$ will not be an integer. But this expression still is useful for investigating average code length. Here we will examine the "ceiling" of this quantity.

Claim 4.1 (Kraft inequality) *Given an alphabet and probabilities $p(A_i)$, define*

$$\lambda_i \equiv \lceil \log \frac{1}{p(A_i)} \rceil .$$

Then,

$$\sum_{i=1}^{N} 2^{-\lambda_i} \ \leq \ 1.$$

Proof

$$\begin{aligned} \lambda_i &\geq \log \frac{1}{p(A_i)} \\ -\lambda_i &\leq \log \ p(A_i) \\ 2^{-\lambda_i} &\leq p(A_i) \end{aligned}$$

Thus,

$$\sum_{i=1}^{N} 2^{-\lambda_i} \ \leq \ 1 \qquad \square$$

Notice that if the $p(A_i)$ is a power of 2 for all i, then we have an equality, not an inequality.

Claim 4.2 *Any prefix code satisfies the Kraft inequality,*

$$\sum_{i=1}^{N} 2^{-\lambda_i} \ \leq \ 1$$

Proof Let $\lambda_{max} = \max\{\lambda_1, \lambda_2, \ldots \lambda_N\}$. For each λ_i, consider a balanced binary tree T_i whose height is $\lambda_{max} - \lambda_i$. This tree has $2^{\lambda_{max}-\lambda_i}$ leaves. Take the prefix code that is given, and extend the binary tree of this code, by replacing each leaf $C(A_i)$ of the prefix code by the tree T_i. All branches of the resulting tree have length $\lambda_{max} = \lambda_i + (\lambda_{max} - \lambda_i)$.

The number of leaves in this new tree is $\sum_{i=1}^{N} 2^{\lambda_{max}-\lambda_i}$. Moreover, since all the leaves in this new tree are at height λ_{max} and since a full binary tree of height λ_{max} has $2^{\lambda_{max}}$ nodes, it must be that

$$\sum_{i=1}^{N} 2^{\lambda_{max}-\lambda_i} \leq 2^{\lambda_{max}}.$$

Dividing both sides by $2^{\lambda_{max}}$ proves the result. $\square$

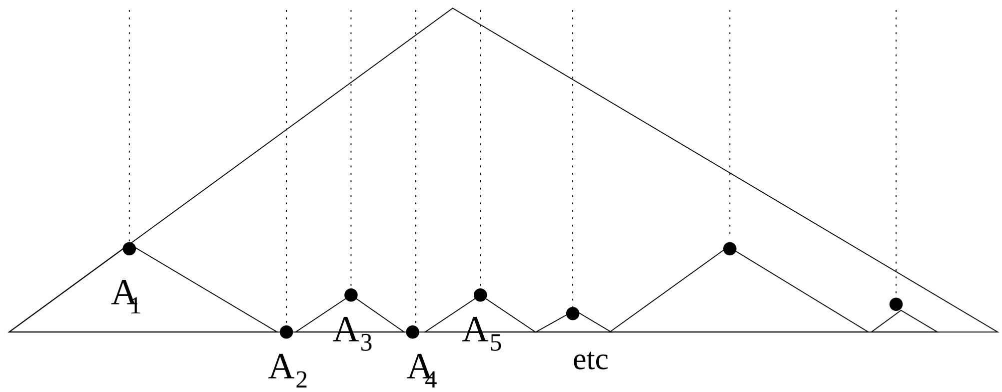
Claim 4.3 (Shannon code) *Define $\lambda_i \equiv \lceil \log \frac{1}{p(A_i)} \rceil$ as before. Then we can construct a prefix code with codeword lengths λ_i. Such a code is called a Shannon code.*

Proof Let λ_{max} be the largest of the λ_i values. Define trees T_i similar to above, so T_i is a balanced binary tree of height $\lambda_{max} - \lambda_i$, with $2^{\lambda_{max}-\lambda_i}$ leaves. In total, these N trees have $\sum_{i=1}^{N} 2^{\lambda_{max}-\lambda_i}$ leaves. Multiplying both sides of the Kraft inequality by $2^{\lambda_{max}}$, we see that

$$\sum_{i=1}^{N} 2^{\lambda_{max}-\lambda_i} \leq 2^{\lambda_{max}}.$$

The left side is the total number of leaves of the T_i and the right side is the number of leaves of a balanced binary tree of height λ_{max}, which we call T.

For each T_i, choose a subtree of T such that leaves of T_i correspond to leaves of T, and no two of these subtrees overlap. (See sketch below. The lengths of the dotted lines represent the λ_i.) Choose our prefix code by assigning the codeword of λ_i to the root of subtree T_i. That is, the codeword $C(A_i)$ is the path from the root of big tree T to the root of little tree T_i, i.e. T_i is embedded in T. This completes the proof. □



Example 1

The Shannon code is not necessarily a good code. For example, take a two symbol alphabet with probabilities .4 and .6. The λ_i defined by the Shannon code are 2 and 1, respectively. But this is clearly not good, since the first codeword doesn't have a sibling and so cannot be optimal.

Entropy

We will next derive upper and lower bounds on average code length. The bounds will be in terms of the following quantity.

Definition 4.1 (Entropy) *Given an alphabet of N symbols and a probability function $p()$ defined on the alphabet, the entropy H is defined as*

$$H \equiv \sum_{i=1}^{N} p(A_i) \log \frac{1}{p(A_i)} .$$

Claim 4.4 *The average code length of a Shannon code satisfies:* $\overline{\lambda} \leq H + 1$.

Proof

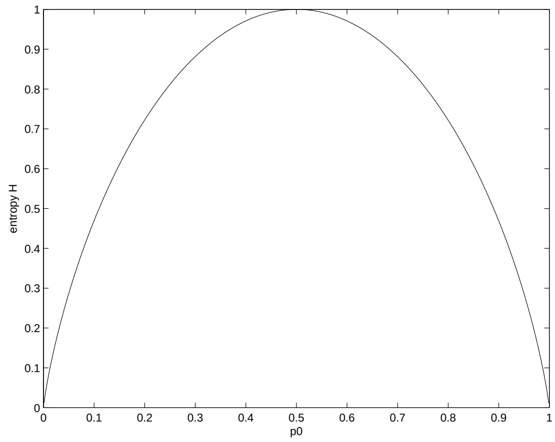
$$\overline{\lambda} = \sum_{i=1}^{N} \lceil \log \frac{1}{p(A_i)} \rceil \, p(A_i) \leq \sum_{i=1}^{N} (\log \frac{1}{p(A_i)} + 1) \, p(A_i) = H + 1$$

Claim 4.5 *The average code length of a Huffman code is bounded above by* $H + 1$.

Proof Since the Huffman code is optimal, its average code length is less than or equal to the average code length of the Shannon code.

Example 2

Consider a two symbol alphabet with probabilities $p(A_1) = p_0$ and $p(A_2) = 1 - p_0$. By definition, the entropy is $p_0 \log \frac{1}{p_0} + (1 - p_0) \log \frac{1}{1-p_0}$. Note that this function is symmetric about $p_0 = \frac{1}{2}$ and that $H = 1$ at $p_0 = \frac{1}{2}$. Verify for yourself that $H \to 0$ when $p_0 \to 0$ (and because of symmetry $H \to 0$ when $p_0 \to 1$). You can verify this by considering the sequence $p_0^{(n)} = \frac{1}{2^n}$.



Example 3

Let $p(A_1) = \frac{1}{2}$, $p(A_2) = \frac{1}{4}$, $p(A_3) = \frac{1}{8}$, $p(A_4) = \frac{1}{8}$.

$$H = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8}$$

Notice that the entropy is the same as average code length of a Huffman code. This is the same example we saw at the beginning of the lecture.