1 Digital Communication

One model for a digital channel is the binary symmetric channel (BSC).

\[ \begin{array}{c}
\text{0} \\
\text{1}
\end{array} \rightarrow \begin{array}{c}
\text{0} \\
\text{1}
\end{array} \]

\[ \begin{array}{c}
\text{1} - P_e \\
\text{P_e}
\end{array} \rightarrow \begin{array}{c}
\text{1} - P_e \\
\text{P_e}
\end{array} \]

Figure 1: Binary Symmetric Channel.

Since a channel causes errors in the transmitted signal, there is usually a source coder between the source and transmitter. Assume the source randomly produces a symbol \( a_n \), where \( a_n \) is in the alphabet \( \{a_1, a_2, ..., a_L\} \).

Each symbol has a probability associated with it. The sum of the probabilities for each symbol is 1.

\[ \sum_{i=1}^{L} Pr(a_i) = 1 \]

One way to measure the randomness of a source is to find the entropy.

**Definition 1.** Entropy:

\[ H(A) = -\sum_{i=1}^{L} Pr(a_i) \log_2(Pr(a_i)) \]

Note that \( H(A) \geq 0 \).

**Example 1.** Find the entropy for alphabet

\[ A = \{a_1, a_2, a_3, a_4\} \]
with probabilities

\[ Pr(a_1) = \frac{1}{2}, \]
\[ Pr(a_2) = \frac{1}{4}, \]
\[ Pr(a_3) = \frac{1}{8}, \]
\[ Pr(a_4) = \frac{1}{8}. \]

The entropy is

\[ H(A) = -\left( \frac{1}{2}(-1) + \frac{1}{4}(-2) + \frac{1}{8}(-3) + \frac{1}{8}(-3) \right), \]
\[ = 1.75 \text{ bits}. \]

**Theorem 1.** Source Coding Theorem: there exists a decodable source code such that \( \bar{B} \), the average number of bits needed, satisfies

\[ H(A) \leq \bar{B} < H(A) + 1. \]

Also, if \( \bar{B} < H(A) \), the code is not decodable.

## 2 Compression

Some examples of compression methods are ZIP, MP3 and JPEG. Compressing using ZIP is decodable, but MP3 and JPEG are both non-decodable compression methods.

Consider the following coding scheme:

\[ a_1 \rightarrow 00 \]
\[ a_2 \rightarrow 01 \]
\[ a_3 \rightarrow 10 \]
\[ a_4 \rightarrow 11 \]

Each letter is coded using two bits, so \( \bar{B} = 2 \). Assuming these letters have the same probabilities as the first example, we have

\[ 1.75 < 2 < 2.75, \]

so the code is decodable. An alternative coding scheme:

\[ a_1 \rightarrow 1 \]
\[ a_2 \rightarrow 01 \]
\[ a_3 \rightarrow 000 \]
\[ a_4 \rightarrow 001 \]

For this coding scheme, \( \bar{B} = 1.75 \). The scheme is optimal!

One method of coding is called **Huffman coding.** To do this, we first order the letters in decreasing order of probabilities. Starting with this values, draw branches combining the two letters with the lowest probabilities. The probability assigned to the end of the branch is the sum of the probabilities. This step is then repeated. After making all the branches, we arbitrarily label them with 0s and 1s. Codewords are made by the branch bit assignment.
Is there a problem with variable length codes? We need to be able to find the boundaries between codewords.

011001111010

The Huffman coding scheme is a prefix code, meaning that no codeword is the prefix of another codeword. For the above bit stream, we can find the boundaries.

0|110|0|0|111|110|10