1 Iterative Decoding

Given a possibly corrupted encoding $c$ of input message $y$, we would like to determine the best estimate of $y$. Recall that the encoding can be generated using the $k \times n$ (Hamming) generator matrix $G$ and errors can be corrected using the $n - k \times n$ (Hamming) parity check matrix $H$.

We can also infer the most likely message using Maximum A Priori estimation (MAP) or Maximum Likelihood Estimation (ML or MLE). MAP is done by maximizing the probability of the code, given the message:

$$\max_{c \in C} P(c|y)$$

whereas ML is done by maximizing the likelihood of the code:

$$\max_{c \in C} P(y|c) = \max_{c \in C} P(y|c) \prod_{i=1}^{n} P(y_i|c_i)$$

In the binary case for a single bit, this becomes:

$$\arg x \in \{0,1\} \max c \in C \max \sum_{c \in C} P(c|y)$$

$$= \arg x \in \{0,1\} \max \sum_{c \in C} P(y|c)$$

$$= \arg x \in \{0,1\} \max \sum_{c \in C} \prod_{i=1}^{n} P(y_i|c_i)$$

We denote the variable degree as $d_v$ and the check degree as $d_c$. Then the transmission rate is $\frac{n \cdot d_v}{d_c}$ where $n$ is the number of bits. This should be clear from the image shown above.

The Iterative Decoding (or Message Passing) algorithm is a two step iterative process used to find the best decoding of the encoded message received. In the first step, called the variable rule, we compute the log-likelihood ratio as follows:
\[ m = \log \frac{P(x = 0|y_1\ldots y_d)}{P(x = 1|y_1\ldots y_d)} \]
\[ = \log \frac{P(y_1\ldots y_d|x = 0)P(x = 0)P(y_1\ldots y_d)}{P(y_1\ldots y_d|x = 1)P(x = 1)P(y_1\ldots y_d)} \]

Since the priors are the same for both 0 and 1, we can cancel \( P(x = 0) \) with \( P(x = 1) \). Then

\[ m = \log \frac{P(y_1\ldots y_d|x = 0)}{P(y_1\ldots y_d|x = 1)} \]
\[ = \log \frac{\prod_{i=1}^{d} P(y_i|x_i = 0)}{\prod_{i=1}^{d} P(y_i|x_i = 1)} \]

The second equivalence follows from the independence assumption of the memoryless channel. Then, simplifying the log of products to a sum of logs, we get

\[ m = \sum_{i=1}^{d} \log \frac{P(y_i|x_i = 0)}{P(y_i|x_i = 1)} \]
\[ = \sum_{i=1}^{d} l_i \]

where \( l_i \) is the log-likelihood ratio for a single bit.

For the second step of the Iterative Decoding algorithm, the check rule, we have \( d = d_c \) and the log-likelihood ratio is

\[ m = \log \frac{P(x = 0|y_1\ldots y_{d-1})}{P(x = 1|y_1\ldots y_{d-1})} \]

Notice the change of the bounds on \( y \). Exponentiating the log-likelihood ratio gives

\[ e^m = \frac{P(x = 0|y_1\ldots y_{d-1})}{P(x = 1|y_1\ldots y_{d-1})} \]

Then it follows that

\[ \frac{e^m - 1}{e^m + 1} = \frac{P(x = 0|y_1\ldots y_{d-1})}{P(x = 1|y_1\ldots y_{d-1})} - 1 \]
\[ \frac{e^m - 1}{e^m + 1} = \frac{P(x = 0|y_1\ldots y_{d-1})}{P(x = 1|y_1\ldots y_{d-1})} + 1 \]

Multiplying both numerator and denominator by \( P(x = 1|y_1\ldots y_{d-1}) \) gives

\[ \frac{e^m - 1}{e^m + 1} = \frac{P(x = 0|y_1\ldots y_{d-1}) - P(x = 1|y_1\ldots y_{d-1})}{P(x = 1|y_1\ldots y_{d-1}) + P(x = 0|y_1\ldots y_{d-1})} \]

The denominator evaluates to 1, because it the sum of a distribution over the entire outcome space.
\[
\frac{e^m - 1}{e^m + 1} = P(x = 0|y_1...y_{d-1}) - P(x = 1|y_1...y_{d-1}) \\
= P(x_1 + x_2 + ... + x_{d-1} = 0|y_1...y_{d-1}) - P(x_1 + x_2 + ... + x_{d-1} = 1|y_1...y_{d-1}) \\
= \prod_{i=1}^{d-1} (P(x_i = 0|y_i) - P(x_i = 1|y_i))
\]

For a single bit, the log-likelihood ratio is:

\[
l_i = \log \frac{P(y_i|x_i = 0)}{P(y_i|x_i = 1)} = \log \frac{P(x_i = 0|y_i)}{P(x_i = 1|y_i)}
\]

Then the ratio

\[
\frac{e^m - 1}{e^m + 1} = P(x_i = 0|y_i) - P(x_i = 1|y_i) \\
= \prod_{i=1}^{d-1} \frac{e^{l_i} - 1}{e^{l_i} + 1} \\
= \prod_{i=1}^{d-1} \frac{e^{l_i} - e^{l_i/2}}{e^{l_i} + e^{l_i/2}} \\
= \prod_{i=1}^{d-1} \tanh\left(\frac{l_i}{2}\right)
\]

Notice that

\[
(e^m - 1) + (e^m + 1) = 2e^m
\]

Dividing everything by \(e^m + 1\) gives the equality:

\[
\frac{e^m - 1}{e^m + 1} + 1 = \frac{2e^m}{e^m + 1}
\]

Likewise, notice that

\[
(e^m + 1) - (e^m - 1) = 2
\]

Dividing by \(e^m + 1\) gives:

\[
1 - \frac{e^m - 1}{e^m + 1} = \frac{2}{e^m + 1}
\]

Then it follows that
\[
e^m = \frac{1 + \prod_{i=1}^{d-1} \tanh(\frac{l_i}{2})}{1 - \prod_{i=1}^{d-1} \tanh(\frac{l_i}{2})}
\]

which implies

\[
m = \log \frac{1 + \prod_{i=1}^{d-1} \tanh(\frac{l_i}{2})}{1 - \prod_{i=1}^{d-1} \tanh(\frac{l_i}{2})}
\]

1.1 Implementation
The algorithm follows directly from the steps above. In each iteration of the message passage algorithm, do the following:

1. Initialization
2. Variable Rule
3. Check Rule

2 Expander Graphs
An expander graph is a strongly connected sparse graph. We can define an expander graph over the encoding as follows

\[
G = \{ S \in V_1 : |S| \leq \alpha n, |N(S)| \geq \gamma d_v |S| \}
\]

for some positive \( \alpha \) and \( \gamma \), where \( V_1 \) is the set of nodes in the message prior to encoding and \( N(S) \) is the set of neighbors of \( S \).

In the next class we will show that if \( \gamma > \frac{3}{4} \), we will be able to correct up to \( \frac{\alpha}{2} n = O(n) \) errors.