This week, we will expand the framework of LDPC codes by studying an efficient message passing decoding algorithm for decoding LDPC codes over general binary input memoryless symmetric (BMS) channels.

1 Channel

A binary memoryless symmetric channel (BMS) is a noisy channel with binary input alphabet (depending on the context we will use either $X = \{+1, -1\}$, or $X = \{0, 1\}$), and channel output $Y \subseteq \mathbb{R}$, satisfying two conditions:

1. The channel output at any given time is conditionally independent on past channel inputs, given the input at the same time.
2. The probability of receiving output $y \in [a, b]$ on input $+1$ is the same as the one of receiving output $y \in [-b, -a]$ on input $-1$.

It turns out that a more general definition is in fact possible, whereby the output alphabet is a general set $Y$ (not necessarily a subset of $\mathbb{R}$).

The symmetry condition can be generalized to this case by assuming that there exist an involution $\phi : Y \to Y$ (i.e. a bijection satisfying $\phi^2(y) = y$ for all $y \in Y$) such that, the probability of receiving $y \in S$ on input $+1$ is equal to the probability of receiving $y \in \phi(S)$ on input $-1$ for any set $S$. Not that the binary erasure channel fits this framework if we let $Y = \{+1, \ast, -1\}$ and $\phi(+1) = -1, \phi(-1) = +1, \phi(\ast) = \ast$.

Formally, the channel is defined by transition probability $\{Q(y | x)\}_{x \in \{\pm 1\}}$, satisfying
\[
Q(y \in \phi(S) | -1) = Q(y \in S | 1),
\]
for any $S \subseteq Y$. Therefore it is sufficient to specify the distribution $Q(\cdot | +1)$ to define the channel.

For any BMS channel, we can construct an equivalent BMS channel with output alphabet $Y \subseteq \mathbb{R}$. This can be done by defining the log-likelihood ratio
\[
\ell(y) = \log \frac{Q(y | +1)}{Q(y | -1)}.
\]
The channel $x \to \ell(y)$ is equivalent to the original one, in the sense that its output contains exactly the same information about the channel input, as the original channel. Technically, $\ell(y)$ is a sufficient statistics for $x$ given $y$. The same statement applies to repeated uses of the channel.

Thanks to this transformation, we will always assume the output alphabet to be $Y \subseteq \mathbb{R}$.

In practice we will mainly be interested in two types of channels. In the first case $Y = \mathbb{R}$ and the transition probability is absolutely continuous with respect to the Lebesgue measure: in this case we will denote the density as $Q(y | x)$. An example is the binary additive white gaussian noise channel BAWGN($\sigma^2$), where we have
\[
Q(y | x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y - x)^2}{2\sigma^2} \right\}.
\]
In the second case $Y$ is finite, and $Q(y | x)$ will denote the probability mass function. A classical example is the binary symmetric channel BSC($\varepsilon$), with $Y = X = \{+1, -1\}$, and
\[
Q(y | x) = \begin{cases} 1 - \varepsilon & \text{if } y = x, \\ \varepsilon & \text{otherwise}. \end{cases}
\]
The capacity of a BMS channel with transition probability \( Q(\cdot | + 1) \) reads (assuming for simplicity the output alphabet to be discrete)

\[
C(Q) = \sum_{y \in Y} Q(y | + 1) \log_2 \frac{2Q(y | + 1)}{Q(y | + 1) + Q(y | - 1)}.
\]

(5)

2 Message Passing

‘Message passing’ decoders are the natural generalization of what we discussed for the erasure channel. They are iterative algorithms, whose basic variables are messages. Given an edge \((i, a) \in E\), with \( i \in V \) a variable node, and \( a \in C \) a check node, the variable-to-check and check-to-variable messages are denoted by \( \nu_{i \rightarrow a} \) and \( \hat{\nu}_{a \rightarrow i} \). We shall omit time labels unless necessary.

The message outgoing from a node at a given time is a function of incoming messages through other edges at the previous time step. After a certain number of iterations, the decision on bit \( x_i \) is taken on the basis of all incoming messages at the same node.

Here are three examples of message passing algorithms.

Gallager A algorithm. Assume transmission to take place over the BSC(\( \varepsilon \)) and denote by \( y_i \in \{+1, -1\} \) the received symbol at position \( i \in V \). Messages take values in \( M = \{+1, -1\} \) and are updated according to the rules

\[
\nu_{i \rightarrow a}^{(t+1)} = \begin{cases} 
+1 & \text{if } \hat{\nu}_{b \rightarrow i}^{(t)} = +1 \text{ for all } b \in \partial i \setminus a, \\
-1 & \text{if } \hat{\nu}_{b \rightarrow i}^{(t)} = -1 \text{ for all } b \in \partial i \setminus a, \\
y_i & \text{otherwise},
\end{cases}
\]

(6)

\[
\hat{\nu}_{a \rightarrow i}^{(t)} = \prod_{j \in \partial a \setminus i} \nu_{j \rightarrow a}^{(t)}.
\]

(7)

How would you generalize the same algorithm to other channel models?

Gallager B algorithm. Again, assume communication over BSC(\( \varepsilon \)) and messages in \( M = \{+1, -1\} \). The update rules are now

\[
\nu_{i \rightarrow a}^{(t+1)} = \text{sign} \left\{ y_i + \sum_{b \in \partial i \setminus a} \hat{\nu}_{b \rightarrow i}^{(t)} \right\},
\]

(8)

\[
\hat{\nu}_{a \rightarrow i}^{(t)} = \prod_{j \in \partial a \setminus i} \nu_{j \rightarrow a}^{(t)}.
\]

(9)

In the first of these equations, ties can be broken in many possible ways. Suggest a few possibilities.

Decoder with erasures. (This has nothing to do with the erasure channel!) We assume communication over the BSC(\( \varepsilon \)), but this time take message set \( M = \{+1, 0, -1\} \). The update equations are formally the same as Eq. (8) and (9), with the convention \( \text{sign}(0) = 0 \).

Belief propagation (BP). Here we consider a general BMS channel, with transition probability \( Q \). Messages are in the set \( M \) of distributions over \( X \equiv \{+1, -1\} \). In other words they are pairs of non-negative real numbers \( (\nu(+1), \nu(-1)) \) with \( \nu(+1) + \nu(-1) = 1 \). In order to write the update equations it is convenient to introduce a notation convention. We will use the symbol \( \equiv \) to indicate identity of probability distributions

\[
\nu_{i \rightarrow a}^{(t+1)} = \text{sign} \left\{ y_i + \sum_{b \in \partial i \setminus a} \hat{\nu}_{b \rightarrow i}^{(t)} \right\},
\]

(8)

\[
\hat{\nu}_{a \rightarrow i}^{(t)} = \prod_{j \in \partial a \setminus i} \nu_{j \rightarrow a}^{(t)}.
\]

(9)
In the case of general decoder, density evolution is a recursive distribution for the distribution of the messages.

3 Density evolution

Density evolution can assume that the all 0 (all +1) codeword has been transmitted.

The update equations can be written as that are executed for the same number of iterations.

\[
\nu^{(t+1)}_{i \rightarrow a}(x_i) \equiv Q(y_i|x_i) \prod_{b \in \partial_i \setminus a} \hat{\nu}^{(t)}_{b \rightarrow i}(x_i),
\]

\[
\hat{\nu}^{(t)}_{a \rightarrow i}(x_i) \equiv \sum_{x_{j(1)} \cdots x_{j(k)}} \hat{\nu}^{(t)}_{j(1) \rightarrow a}(x_{j(1)}) \cdots \hat{\nu}^{(t)}_{j(k) \rightarrow a}(x_{j(k)}) \mathbb{I}(x_i \oplus x_{j(1)} \oplus \cdots \oplus x_{j(k)} = 0).
\]

Where, in the last equation, \( \{1, j(1), \ldots, j(k)\} \equiv \partial a \). Decision at a variable node \( i \) is computed from the belief

\[
\nu^{(t+1)}_{i}(x_i) \equiv Q(y_i|x_i) \prod_{b \in \partial i} \hat{\nu}^{(t)}_{b \rightarrow i}(x_i).
\]

These updates can be justified by noting that, if the factor graph \( G \) is a tree, then after a number of iterations \( t \geq \text{diam}(G) \), then the belief \( \nu^{(t+1)}_{i}(x_i) \) coincides with the posterior distribution of bit \( x_i \), given observations \( \{y_j\}_{j \in V} \).

If the graph is not locally tree-like, then belief propagation is the optimal message passing algorithm. Namely, for any \( t \) such that \( B_G(i; t) \) is a tree, BP computes the posterior distribution of \( x_i \), given the observations \( \{y_j\}_{j \in B_G(i; t)} \). As a consequence, it achieves the lowest bit error rate among all message passing algorithms that are executed for the same number of iterations.

Belief propagation is conveniently written in terms of the log-likelihood ratios which we denote by

\[
h^t_{i \rightarrow a} = \log \frac{\nu^{(t)}_{i \rightarrow a}(+1)}{\nu^{(t)}_{i \rightarrow a}(-1)}, \quad \hat{h}^t_{a \rightarrow i} = \log \frac{\hat{\nu}^{(t)}_{a \rightarrow i}(+1)}{\hat{\nu}^{(t)}_{a \rightarrow i}(-1)}, \quad h^t_i = \log \frac{\nu^{(t)}_{i}(+1)}{\nu^{(t)}_{i}(-1)},
\]

The update equations can be written as

\[
h^{t+1}_{i \rightarrow a} = \ell(y_i) + \sum_{b \in \partial i \setminus a} \hat{h}^t_{b \rightarrow i},
\]

\[
\hat{h}^t_{a \rightarrow i} = 2 \text{atanh} \left\{ \prod_{j \in \partial a \setminus i} \text{tanh} \left( \frac{\hat{h}^t_{j \rightarrow a}}{2} \right) \right\}.
\]

All of the above algorithms are (in some sense) ‘symmetric.’ Argue that, for the sake of the analysis, we can assume that the all 0 (all +1) codeword has been transmitted.

3 Density evolution

In the case of general decoder, density evolution is a recursive distribution for the distribution of the messages.

Consider a general message passing algorithm defined by update rules of the form

\[
\nu^{(t+1)}_{i \rightarrow a} = \Phi(y_i; \{\hat{\nu}^{(t)}_{b \rightarrow i} : b \in \partial i \setminus a\}),
\]

\[
\hat{\nu}^{(t)}_{a \rightarrow i} = \Psi(\{\nu^{(t)}_{j \rightarrow a} : j \in \partial a \setminus i\}).
\]

Density evolution for the same algorithm is defined by the distributional recursion

\[
\nu^{(t+1)} = \Phi(y_i; \hat{\nu}^{(t)}_{1}, \ldots, \hat{\nu}^{(t)}_{k-1}),
\]

\[
\hat{\nu}^{(t)} = \Psi(\nu^{(t)}_{1}, \ldots, \nu^{(t)}_{k-1}).
\]
Despite the formal similarity between these equations and Eqs. (16), (17), one has to pay attention to the difference. In the density evolution \( d \) denoted equality in distribution between random variables. In these equalities, \( \nu(t), \nu_1(t), \nu_2(t), \ldots \) are understood to be iid random variables, and the same for \( \hat{\nu}(t), \hat{\nu}_1(t), \hat{\nu}_2(t), \ldots \). Further \( y \) is distributed according to the transition probability \( Q(y|+1) \), \( l \) with the variable node degree distribution \( \lambda \), and \( k \) with the check node degree distribution \( \rho \).

A more explicit expression can be written by introducing the distributions of \( \nu(t) \) and \( \hat{\nu}(t) \) to be denoted respectively as \( a_t(\cdot) \), \( \hat{a}(\cdot) \). Assuming, for the sake of simplicity that the message alphabet is finite, we can write the density evolution equations as

\[
a_{t+1}(\nu) = \sum_l \lambda_l \sum_y Q(y|+1) \sum_{\hat{\nu} \cdots \hat{\nu}_{l-1}} \hat{a}_t(\hat{\nu}_1) \cdots \hat{a}_t(\hat{\nu}_{l-1}) I\{\nu = \Phi(y; \hat{\nu}_1, \ldots, \hat{\nu}_{l-1})\}, \quad (20)
\]

\[
\hat{a}_t(\hat{\nu}) = \sum_k \rho_k \sum_{\nu_1 \cdots \nu_{k-1}} a_t(\nu_1) \cdots a_t(\nu_{k-1}) I\{\hat{\nu} = \Psi(\nu_1, \ldots, \nu_{k-1})\}. \quad (21)
\]

A prominent application is the analysis of BP decoding. Starting from the LLR updates (14), (15), we get the following density evolution recursion

\[
h_{t+1} d = \ell(y) + \sum_{l=1}^{k-1} \hat{h}_l, \quad (22)
\]

\[
\hat{h}_t d = 2\text{atanh} \left\{ \prod_{j=1}^{l-1} \tanh \left( \frac{\hat{h}_t}{2} \right) \right\}. \quad (23)
\]

The first equation corresponds to a convolution of densities. The second is a convolution in \( \log |\tanh(h/2)| \) domain.

**Summary**

At the end of this week you should know:

1. How LDPC codes are used to communicate over general BMS channels.
2. How to define and simulate message passing algorithms.
3. Definition and basic properties of belief propagation.

This material can be found in Section 4.1-4.3 of MCT page 175 and afterwards. Additional material on BP is in Chapters 14, 15 of the book by M. Mézard and myself, ‘Information, Physics, and Computation’.

**Homework**

Write a program to simulate BP decoding of an irregular LDPC code used over the binary symmetric channel. Use it to plot performance curves (bit error probability) for the regular ensemble \( L(x) = x^3, R(x) = x^6 \), and for a few blocklengths of your choice.

I expect to receive

1. A print-out of the code.
2. The values of the blocklengths \( n \) you used. A description of how you got rid of double edges.
3. A plot of the bit error probability curves.