This week, we will expand the framework of LDPC codes in two directions:

- We will show how density evolution for the BEC extends naturally to irregular ensembles. It turns out the the added degree of freedom provided by the degree distributions allows to achieve the channel capacity in this case, as proved by Luby and collaborators. [We will not prove this fact, because spatial coupling provides another –more general– method to achieve capacity.]
- We will construct an efficient message passing decoding algorithm for decoding LDPC codes over general binary input memoryless symmetric (BMS) channels.

### Density evolution for irregular ensembles

Consider a random Tanner graph with (node perspective) degree distributions $L, R$. As $n \to \infty$, the neighborhood $B_i(t)$ of a uniformly random vertex $i$ converges to a random rooted tree $T_{\lambda, \rho}(t)$. This is a bipartite tree with offspring distributions $\lambda$, for variable nodes, and $\rho$, for check nodes. More precisely, the probability for a variable node to have $i-1$ descendants (and thus $i$ neighbors) is $\lambda_i$, while for a check node is $\rho_i$.

The density evolution equation in this case reads

$$x_{t+1} = \epsilon \lambda(1 - \rho(1 - x_t)),$$

with initial condition $x_0 = 1$. This recursion can also be written as $\lambda^{-1}(x_{t+1}/\epsilon) = 1 - \rho(1 - x_t)$. Show that

$$\int_0^1 \epsilon \lambda(x) \, dx = \frac{\epsilon}{L'(1)}, \quad \int_0^1 [1 - \rho(1 - x)] \, dx = 1 - \frac{1}{R'(1)}.$$  

What does this mean? What does this imply about capacity achieving degree distributions (degree sequences)?

### Channel

A binary memoryless symmetric channel (BMS) is a noisy channel with binary input alphabet (depending on the context we will use either $\mathcal{X} = \{+1, -1\}$, or $\mathcal{X} = \{0, 1\}$), and channel output $y \in \mathcal{Y} \subseteq \mathbb{R}$, 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 $\mathcal{Y}$ (not necessarily a subset of $\mathbb{R}$). How can the symmetry condition be generalized to this case? Show that the binary erasure channel fits this framework.

Formally, the channel is defined by transition probability $\{Q(\cdot | x)\}_{x \in \{\pm 1\}}$, satisfying

$$Q(y \in [-b, -a] | -1) = Q(y \in [a, b] | 1).$$
Therefore it is sufficient to specify the distribution $Q(\cdot | + 1)$ to define the channel. Write the capacity of a BMS channel in terms of $Q(\cdot | + 1)$.

In practice we will mainly be interested in two types of channels. In the first case $\mathcal{Y} = \mathbb{R}$ 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

$$Q(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y-x)^2}{2\sigma^2} \right\}. \quad (4)$$

In the second case $\mathcal{Y}$ is finite, and $Q(y|x)$ will denote the probability mass function. A classical example is the binary symmetric channel BSC($\epsilon$), with $\mathcal{Y} = \mathcal{X} = \{+1, -1\}$, and

$$Q(y|x) = \begin{cases} 1 - \epsilon & \text{if } y = x, \\ \epsilon & \text{otherwise.} \end{cases} \quad (5)$$

**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($\epsilon$) 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

$$\nu^{(t+1)}_{i \rightarrow a} = \begin{cases} +1 & \text{if } \hat{\nu}^{(t)}_{b \rightarrow i} = +1 \text{ for all } b \in \partial i \setminus a, \\ -1 & \text{if } \hat{\nu}^{(t)}_{b \rightarrow i} = -1 \text{ for all } b \in \partial i \setminus a, \\ y_i & \text{otherwise,} \end{cases} \quad (6)$$

$$\hat{\nu}^{(t)}_{a \rightarrow i} = \prod_{j \in \partial a \setminus i} \nu^{(t)}_{j \rightarrow a}. \quad (7)$$

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

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

$$\nu^{(t+1)}_{i \rightarrow a} = \text{sign} \left\{ y_i + \sum_{b \in \partial i \setminus a} \hat{\nu}^{(t)}_{b \rightarrow i} \right\}, \quad (8)$$

$$\hat{\nu}^{(t)}_{a \rightarrow i} = \prod_{j \in \partial a \setminus i} \nu^{(t)}_{j \rightarrow a}. \quad (9)$$

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($\epsilon$), 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 $\text{sign}(0) = 0$. 

2
Belief propagation (BP). Here we consider a general BMS channel, with transition probability $Q$. Messages are in the set $\mathcal{M}$ of distributions over $\mathcal{X} \equiv \{+1, -1\}$. 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 ‘up to a normalization.’ Then we have the update rules

$$
\begin{align*}
\nu_{i \rightarrow a}^{(t+1)}(x_i) & \equiv Q(y_i|x_i) \prod_{b \in \partial \setminus a} \hat{\nu}_{b \rightarrow i}^{(t)}(x_i), \\
\hat{\nu}_{a \rightarrow i}^{(t)}(x_i) & \equiv \sum_{x_j(1) \ldots x_j(k)} \nu_j^{(t)}(x_j(1)) \ldots \nu_j^{(t)}(x_j(1)) I(x_i \oplus x_j(1) \oplus \cdots \oplus x_j(k) = 0).
\end{align*}
$$

(10)

(11)

Where, in the last equation, $\{1, j(1), \ldots, j(k)\} \equiv \partial a$. What is the origin of these equations?

Belief propagation is (in some sense) the best message passing algorithm. Can you explain this statement, and why is it correct?

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.

LLR’s

The channel output $y$ on input $x \in \{+1, -1\}$ is conveniently represented in term of log-likelihood ratios

$$
\ell(y) = \log \frac{Q(y|+1)}{Q(y|-1)}.
$$

(12)

The same representation can be used for BP. Write the BP update equations in terms of log-likelihood ratios.

SUMMARY

At the end of this week you should know:

1. How to analyze standard irregular ensembles on the erasure channel through density evolution.
2. How LDPC codes are used to communicate over general BMS channels.
3. How to define and simulate message passing algorithms.
4. 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’.

WORK

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 ensemble $L(x) = 0.5 x^2 + 0.3 x^3 + 0.2 x^{10}$, $R(x) = 0.7 x^7 + 0.3 x^8$, 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.