1 Symmetry, monotonicity and thresholds

We say that a random variable $Z$ is symmetric if, for any bounded measurable function $f : \mathbb{R} \to \mathbb{R}$, the following holds:

$$
\mathbb{E}\{f(-Z)\} = \mathbb{E}\{e^{-Z} f(Z)\} .
$$

A probability measure over $\mathbb{R}$ is symmetric, if it is the law of a symmetric random variable. In order to prove that $Z$ (or its law) is symmetric (and assuming it tail probability decay fast enough as to make the expressions below well defined) it is sufficient to check it for functions of the form $f(x) = e^{\lambda x}$. In other words, it is sufficient to check that, for all $\lambda \in \mathbb{R}$,

$$
\mathbb{E}\{e^{-\lambda Z}\} = \mathbb{E}\{e^{-(1-\lambda)Z}\} .
$$

In other words the moment generating functions $\lambda \mapsto F(p) = \mathbb{E}\{e^{\lambda Z}\}$ is symmetric around $\lambda = 1/2$.

If $\{Q(y|x)\}_{x \in \{+1,-1\}, y \in \mathcal{Y}}$ is the transition probability of a BMS channel, then the log-likelihood

$$
\ell(Y) = \log \frac{Q(Y|+1)}{Q(Y|-1)} , \quad Y \sim Q(\cdot | +1)
$$

is a symmetric random variable. In order to prove this claim, we use the characterization. Let $F(\lambda) = \mathbb{E}\{e^{-\lambda \ell(Y)}\}$. Then for the sake of simplicity, we treat the output alphabet as discrete:

$$
F(\lambda) = \sum_y Q(y|+1) \left[ \frac{Q(y|+1)}{Q(y|-1)} \right]^{-\lambda} = \sum_y Q(y|+1)^{1-\lambda} Q(y|-1)^{\lambda}
$$

By using the channels symmetry, it is easy to see that $F(\lambda) = F(1-\lambda)$, i.e. $\ell(Y)$ is a symmetric random variable.

Viceversa, if $P$ is the probability distribution of a symmetric random variable (which we can assume, for the sake of simplicity, to take values in a finite set $\mathcal{Y} \subseteq \mathbb{R}$), then we can define a BMS channel via

$$
Q(y|+1) \equiv P(y) , \quad Q(y|-1) \equiv P(-y) .
$$

Note that the symmetry property implies that the channel output coincides with the log-likelihood ratio $\ell(y) = y$. By this correspondence, symmetric densities admit a partial ordering by physical degradation $Z_1 \leq Z_2$ if the corresponding channels are ordered in the same way $Q_1 \leq Q_2$ (i.e. $Q_1$ is less noisy than $Q_2$). We write $Q_1 \leq Q_2$ if there exists a third channel $Q$ such that $Q_2$ is equal to $Q_1$ concatenated with $Q$.

Consider now belief propagation decoding with messages $\{h_{i \rightarrow a}^t\}$, $\{\hat{h}_{a \rightarrow i}^t\}$ and soft decisions $\{h_i^t\}$, all of these in the log-likelihood ratio domain. We saw that as $n \to \infty$– these converge to limit random variables described by density evolution

$$
\begin{align*}
h_{i \rightarrow a}^t & \xrightarrow{d} h^t , \\
\hat{h}_{a \rightarrow i}^t & \xrightarrow{d} \hat{h}^t , \\
& h_i^t \xrightarrow{d} h_i^* .
\end{align*}
$$
The random variables \( h^t, \hat{h}^t, h^*_t \) are symmetric. This can be proved in two different ways: (1) By induction over the number of iteration \( t \), using the density evolution recursion; (2) Showing that each of these random variables is the LLR for a certain BMS channel. For instance, \( h^*_t \) correspond to the channel described by the following process. Generate \( x_i \in \{+1,-1\} \) uniformly at random. Generate a rooted bipartite graph of depth \( t \), according to the distribution of local neighborhoods in the code. Generate a codeword for this neighborhood, conditional on \( x_i \). Transmit all the bits in this neighborhood through the original BMS.

Consider a family \( \{\text{BMS}(\varepsilon)\}_{\varepsilon > 0} \) ordered by physical degradation with respect to the parameter \( \varepsilon \). The asymptotic bit error rate after \( t \) iterations can be written as

\[
\mathbb{P}_b(t; \varepsilon) = \mathbb{P}(h^t_u < 0) + \frac{1}{2} \mathbb{P}(h^t_u = 0),
\]

where we assume that the decision on bit \( x_i \) is taken uniformly at random when \( h^t_i = 0 \). Using the symmetry property this can be also written as

\[
\mathbb{P}_b(t; \varepsilon) = \mathbb{E}\{f(h^t_u)\},
\]

\[
f(x) = \begin{cases} e^{-x/2} & \text{if } x \geq 0, \\ 1/2 & \text{if } x < 0. \end{cases}
\]

Notice that it follows from the ordering of the channel and the optimality of BP that \( \mathbb{P}_b(t; \varepsilon) \) is non-increasing in \( t \), and non-decreasing in \( \varepsilon \). In particular, we can define the limit curve

\[
\lim_{t \to \infty} \mathbb{P}_b(t; \varepsilon) = \mathbb{P}_b(\varepsilon).
\]

The curve \( \varepsilon \mapsto \mathbb{P}_b(\varepsilon) \) is non-decreasing. The BP threshold is defined by

\[
\varepsilon_{\text{BP}} = \sup \left\{ \varepsilon \geq 0 : \mathbb{P}_b(\varepsilon) = 0 \right\}.
\]

(13)

\[
= \sup \left\{ \varepsilon \geq 0 : \lim_{t \to \infty} \mathbb{P}_b(\varepsilon; t) = 0 \right\}.
\]

(14)

\[
= \sup \left\{ \varepsilon \geq 0 : \text{Law}(h^t) \to \delta_{\varepsilon} \right\}.
\]

(15)

The second identity follows from Eq. (10).

Thresholds can be estimated by implementing density evolution numerically. We discussed in class a specific approach that is usually referred to as ‘sampled density evolution’ or ‘population dynamics’.

\section{The area theorem}

There is one metric that is more fundamental than bit error rate. In the case of communication over the erasure channel this metric is known as the EXIT (extrinsic information) function. For other BMS channel, it is known as GEXIT function.

Given a codeword \( X \in \{+1,-1\}^n \), transmitted through channel \( \text{BEC}(\varepsilon) \), and channel output \( Y \in \{+1,-1,\ast\}^n \), we define

\[
\text{EXIT}_n(\varepsilon) = \frac{1}{n} \sum_{i=1}^n H(X_i | Y_{\sim i}),
\]

where \( Y_{\sim i} = (Y_j)_{j \in [n]\setminus i} \). We then have the useful identity

\[
\frac{1}{n} \frac{d}{d\varepsilon} H(X | Y(\varepsilon)) = \text{EXIT}_n(\varepsilon),
\]

(17)
first put forward by Ashikhmin, Kramer and ten Brink. The simplest way to prove this fact is to note that

$$H(X|Y) = H(X_i|Y) + H(X|X_i, Y).$$  \hfill (18)

Imagine that the $i$-th bit is transmitted through a channel with parameter $\varepsilon_i$. Summing over all possible values of $Y_i$, we have

$$H(X_i|Y) = (1 - \varepsilon_i)H(X_i|Y_i \in \{0, 1\}, Y_{\sim i}) + \varepsilon_i H(X_i|Y_i = *, Y_{\sim i})$$  \hfill (19)

$$= \varepsilon_i H(X_i|Y_{\sim i}).$$  \hfill (20)

Differentiating with respect to $\varepsilon_i$,

$$\frac{\partial H(X|Y)}{\partial \varepsilon_i} = \frac{\partial H(X_i|Y)}{\partial \varepsilon_i} = H(X_i|Y_i).$$  \hfill (21)

A similar identity holds for a general BMS($\varepsilon$) channel

$$\frac{1}{n} \frac{dH(X|Y)}{d\varepsilon} = \text{GEXIT}_n(\varepsilon).$$  \hfill (22)

For a channel with transition probability $P(y|x)_{x \in \{+1, -1\}, y \in \mathcal{Y}}$, with derivative with respect to $\varepsilon$ given by $Q$, we have

$$\text{GEXIT}_n(\varepsilon) = \frac{1}{n} \sum_{i=1}^{n} g_i(\varepsilon),$$  \hfill (23)

$$g_i(\varepsilon) = \sum_{y \in \mathcal{Y}} E \left\{ \hat{Q}(y) + 1 \log \left( \sum_{x \in \{+1, 1\}} \frac{P(X_i = x|Y_{\sim i})Q(y|x)}{P(X_i = 0|Y_{\sim i})Q(y|0)} \right) \right\}.$$  \hfill (24)

(We assume communication using a linear code.)

This can be further simplified by writing it in terms of the extrinsic log-likelihood ratio at vertex $i$, defined as

$$h_i^{\text{ext}} = \log \frac{P(X_i = 1|Y_{\sim i})}{P(X_i = 0|Y_{\sim i})},$$  \hfill (25)

$$g_i(\varepsilon) = E[\mathcal{F}(h_i^{\text{ext}})],$$  \hfill (26)

where the kernel $\mathcal{F}$ depends on the channel transition probability, namely

$$\mathcal{F}(z) = \sum_{y \in \mathcal{Y}} \hat{Q}(y) + 1 \log_2 \left( 1 + \frac{Q(y|-1)}{Q(y|+1)} e^{-z} \right).$$  \hfill (27)

The EXIT and GEXIT curves can be upper bounded by the corresponding BP curves, which are obtained by replacing mean over the actual LLR with the mean over the fixed point density achieved by density evolution.

$$\text{GEXIT}_n(\varepsilon) \leq \text{GEXIT}_{\text{BP}}(\varepsilon),$$  \hfill (28)

$$\text{GEXIT}_{\text{BP}}(\varepsilon) = \int \mathcal{F}(z) \tilde{a}_z(dz).$$  \hfill (29)

Here $\tilde{a}_t$ is the distribution of

$$\tilde{h}^{t+1} \overset{d}{=} \sum_{b=1}^{k} \tilde{h}_b,$$  \hfill (30)

where $k \sim L$.  

3
Protograph codes were introduced in (Thorpe, 2003) and introduce some useful design freedom. They are also a good intermediate step towards spatially coupled codes.

Let \( G = (V, C, E) \) be a factor graph with variable nodes \( V \), check nodes \( C \), and edges \( E \). Here and in the following, we admit multiple edges, i.e. \( G \) is a multigraph. We say that \( G' = (V', C', E') \) is an \( m \)-fold lift of \( G \) if there exist a surjective map \( \pi : V' \rightarrow V \) and \( \pi : C' \rightarrow C \), such that (calling \( N(i) \) the neighborhood of vertex \( i \)):

- For any \( \pi(i) \in V \), \( \pi(C') = C \).
- For any \( i \in V \cup C \), \(|\pi^{-1}(i)| = m \).
- For any \( i \in V' \cup C' \), \( \pi \) is a graph isomorphism between \( N(i) \) and \( N(\pi(i)) \).

Concretely, we can construct an \( m \)-lift by letting

\[
\begin{align*}
V' &= V \times \{1, 2, \ldots, m\}, \\
C' &= C \times \{1, 2, \ldots, m\},
\end{align*}
\]

and letting, for each \( e \in E \), \( \sigma_e \) be a permutation over \( m \) objects. We then construct the edge set \( E' \) as follows. Given \( (a, k) \in C' \) (with \( a \in C \), \( k \in [m] \)), and \( (i, q) \in V' \) (with \( i \in V \), \( q \in [m] \)), such that \( (a, i) \in E \), we let

\[
(a, k), (i, q) \in E' \iff \sigma_{(a,i)}(k) = q.
\]

\( G' \) is a random protograph if each \( \sigma_e \) is uniformly random.

Figure 1 shows how the classical \((3, 6)\) code can be viewed as a random protograph code.

Protograph codes can be used to enforce some useful structures in LDPC ensembles. For instance, we might want to introduce degree-2 variable nodes but without having loops formed by such nodes. Figure 1 shows an example of a protograph enforcing this constraint.

### 4 Density evolution for protograph codes

Protograph codes can be analyzed using density evolution. This holds asymptotically as \( L \rightarrow \infty \) for \( G \) a fixed (small) graph.

The analysis is somewhat more complex than in the classical ensembles. It is convenient to introduce some notations:

- Given probability distributions \( \hat{a}_1, \ldots, \hat{a}_l \) on \( \mathbb{R} \), we denote by \( \hat{a}_1 \otimes \cdots \otimes \hat{a}_l \) their convolution, i.e. the distribution of \( x_1 + \cdots + x_l \), when \( x_1 \sim \hat{a}_1, \ldots, x_l \sim \hat{a}_l \).
Given probability distributions $a_1, \ldots, a_k$ on $\mathbb{R}$, we denote by $a_1 \boxoplus \cdots \boxoplus a_l$ the distribution of

$$2 \text{atanh}(\tanh(x_1/2) \cdots \tanh(x_k/2)),$$

when $x_1 \sim a_1, \ldots, x_k \sim a_l$.

We denote by $c$ the distribution of $\ell(y)$ when $y \sim Q(| \cdot | + 1)$.

With these notations, sensitivity evolution for Gallager’s $(l, k)$ ensemble reads

$$a_{l+1} = c \boxoplus \hat{a}_l^{(l-1)},$$

$$\hat{a}_l = a_l^{(l-1)}.$$  \hfill (35)

In a protograph code, we need to keep track of one density per each directed edge of the protograph, namely $(a_{i \rightarrow a}, \hat{a}_{i \rightarrow a})_{(a,i) \in E}$. These are updated according to the rules

$$\hat{a}_{i \rightarrow a}^{t+1} = c \boxoplus \left( \bigoplus_{b \in E_{i \rightarrow a}} \hat{a}_b^{t} \right),$$

$$\hat{a}_{i \rightarrow a}^{t} = \hat{a}_{i \rightarrow a}^{(t-1)}.$$  \hfill (36)

For instance in the case of decoding for the erasure channel, these densities reduce to a single number: $a_{i \rightarrow a}^{t} = (1 - z_{i \rightarrow a}^{t}) \delta_0 + z_{i \rightarrow a}^{t} \delta_b$, $\hat{a}_{i \rightarrow a}^{t} = (1 - \hat{z}_{i \rightarrow a}^{t}) \delta_0 + \hat{z}_{i \rightarrow a}^{t} \delta_b$. These are updated using

$$z_{i \rightarrow a}^{t+1} = \bar{z}_{i \rightarrow a}^{t+1},$$

$$z_{i \rightarrow a}^{t} = 1 - \prod_{j \in E_{i \rightarrow a}} (1 - \hat{z}_{j \rightarrow a}^{t}).$$  \hfill (40)

5 Spatially coupled ensembles

Spatially coupled codes are a class of LDPC codes that can be proved to achieve capacity over BMS channels in a universal manner. This means that the code construction does not depend on the channel, but only on the rate $r$. The code will then allow reliable communication under BP decoding for any BMS channel with capacity $C > r$ (as the blocklength diverges). Notice that the decoder depends on the channel.

The code construction is more complicated than for standard LDPC ensemble and involves several design parameters, and hence it is useful to first give a high-level description of the steps involved. Thought we will assume we are communicating over a BMS($\varepsilon$) channel (a family ordered by physical degradation), with capacity $C_{\text{BMS}}(\varepsilon)$

1. Start with a regular $(k, l)$ ensemble. Its design rate is $r_0 = 1 - (l/k)$, and will be the asymptotic rate achieved by the spatially coupled code. We denote by $\varepsilon_{\text{MAP}}(k, l)$ the MAP (maximum a posteriori probability) threshold for that code. This corresponds to computing for each bit, the posterior distribution $P(x_i|y_1, \ldots, y_n)$ and using that for decoding.

2. Construct a ‘spatially coupled’ protograph $G_L$, based on that the $(k, l)$ ensemble. This is the core of the construction and depends on an additional parameter $L \in \mathbb{N}$, which will be described below. We denote by $\varepsilon_{\text{BP}}(k, l; L)$ the BP threshold for the protograph code with protograph $G_L$.

3. Lift $G_L$ $m$ times to construct a random $m$-lift $G_L(m)$. This will be the actual factor graph of our code.

The BP threshold of code $G_L(m)$ is $\varepsilon_{\text{BP}}(k, l; L)$ and can be evaluated – for fixed $(k, l, L)$ – by density evolution, as explained in the previous section. The key technical result is the following threshold saturation phenomenon.
Figure 2: Illustration of threshold saturation.

**Theorem 1** (Kudekar, Richardson, Urbanke, 2012).

$$\lim_{L \to \infty} \varepsilon_{BP}(k, l; L) = \varepsilon_{MAP}(k, l).$$  \hfill (41)

This implies that by letting $k, l$ be large enough, we can achieve reliable communication at any noise level below the Shannon threshold $\varepsilon_{Shannon}(r) = \sup \{\varepsilon : C(\varepsilon) > r\}$. This happens because of the following classical result.

**Theorem 2** (Gallager, 1962). The MAP threshold of an $(l, k)$ LDPC ensemble converges to the Shannon threshold as $l, k \to \infty$ with $1 - l/k \to r$. Namely

$$\lim_{k \to \infty} \varepsilon_{MAP}(k, l = [k(1 - r)]) = \varepsilon_{Shannon}(r).$$  \hfill (42)

The last theorem is not entirely surprising, since we already saw that high-density parity check codes achieve capacity. It is however surprising that it is sufficient to use a number of ones going to infinity at any rate with $n$. In fact the convergence in the last theorem is exponentially fast in $k$, and therefore it is sufficient to use moderate values of $(k, l)$ to get very close to capacity. For instance, on the BSC($\varepsilon$), we have $\varepsilon_{MAP}(6, 3) \approx 0.101, \varepsilon_{MAP}(8, 4) \approx 0.107, \varepsilon_{MAP}(10, 5) \approx 0.108$, to be compared with $\varepsilon_{Shannon}(r = 1/2) \approx 0.1100279$.

Summarizing, we are taking three limits in the following order: (1) $m \to \infty$ (to be able to get the density evolution thresholds); (2) $L \to \infty$ (to get threshold saturation $\varepsilon_{BP}(k, l; L) \to \varepsilon_{MAP}(k, l)$); (3) $l, k \to \infty$ (to achieve capacity $\varepsilon_{MAP}(k, l) \to \varepsilon_{Shannon}(r)$).

We are left with the task of explaining what spatial coupling is about! The idea is to construct a protograph with regular degrees $(k, l)$ for most of the nodes, and whose nodes are arranged along a line. This one-dimensional structure constrains the connectivity of this factor graph: nodes are only connected to other ones that are close on the line. The parameter $L$ corresponds to the length of this line.

There are many ways of defining the details of this constructions, and these do not matter too much. They are typically chosen to make calculations easier or for some implementation reason. Figure 3 provides...
Figure 3: Construction a spatially coupled code.

Figure 4: An equivalent description of the code of Figure 3.

an example of such a construction. We start from the protograph of a \((k,l)\) code, repeat it \(L\) times, arrange the copies on a line, and finally ‘spread edges’ as to couple nearby graphs.

A few remarks are in order:

1. The code of figure 3 has rate which is slightly smaller than the one of the base regular code. However, it is easy to see that, letting \(r = 1 - (l/k)\), we have

\[
r(L) = 1 - \frac{L + l - 1}{L}(1 - r) = r + O(1/L).
\]

and therefore we approach the desired rate as \(L \to \infty\).

2. Bits at the boundary are better protected than the ones in the center of the line structure. Indeed, the corresponding parity checks have smaller degree. An alternative way of seeing this is to look at the same code as a portion of an infinite code in which most bits are set to 0.

Let us emphasize that this is not an accident, but is a crucial feature of the spatial coupling construction, and is of course related to the \(O(1/L)\) rate loss at the previous point. Without this feature, spatial coupling does not work.
3. The construction of figure 3 couples each \((k,l)\) protograph with its immediate neighbors. Strictly speaking, this does not achieve threshold saturation although the gap is so small that it is very difficult to see it in any simulation. In order to achieve threshold saturation, it is necessary to couple neighbors on the line up to some distance \(\ell(L) = o(L)\) which also diverges (arbitrarily slowly) with \(L\).

Again, there are various ways of doing this, but this is a technical requirement which is important mainly in order to prove Theorem 1. We will therefore not discuss it in detail.

The last point to explain is: what is the origin of threshold saturation?

The short answer is that you can do density evolution analysis for the spatially coupled protograph, and observe that, as \(L \to \infty\) the BP threshold improves and saturates to the MAP threshold of the underlying base code. This however does not begin to explain the phenomenon, and does not go a long way towards proving it.

Both numerical experiments and the theory show instead a very interesting behavior for \(\varepsilon_{\text{BP}}(k,l) < \varepsilon < \varepsilon_{\text{MAP}}(k,l)\). As mentioned above, bits ‘near the boundary’ are better protected, and therefore are correctly decoded first (in a sense, they are below the BP threshold). This is equivalent to reducing the length \(L\), and producing a new boundary. Again, bits near the boundary get decoded, and so on. Figure 5 provides a qualitative picture of how BP decoding evolves in a spatially coupled code. The key to the proof is to trace the evolution of this profile.

Summary

At the end of this week you should know:

1. How to approximate numerically density evolution.
2. Basic properties of density evolution.
3. Area theorems.
5. Construction of spatially coupled LDPC codes

The material treated this week can be found in Chapters 3 and 4 of MCT.
Homework

Consider communication over the binary symmetric channel BSC(\(\varepsilon\)). Using an LDPC graph from the (3,6) ensemble. The objective of this homework is to gain some familiarity with density evolution (DE) by implementing sampled DE.

In sampled DE, we replace the distributions \(a(\cdot), \hat{a}(\cdot)\) by samples

\[ \mathcal{P}_t = \{\nu_1, \nu_2, \ldots, \nu_N\}, \quad \hat{\mathcal{P}}_t = \{\hat{\nu}_1, \hat{\nu}_2, \ldots, \hat{\nu}_N\}. \]

These should be thought of as (ideally) iid samples with distribution (respectively) \(a(\cdot), \hat{a}(\cdot)\). At each time step one generates a new sample \(\hat{\mathcal{P}}_t\) from \(\mathcal{P}_t\), and a new \(\mathcal{P}_{t+1}\) from \(\hat{\mathcal{P}}_t\), mimicking the density evolution recursion.

For instance, to generate the sample \(\hat{\mathcal{P}}_t\), one repeats \(N\) times the following operation. Draw an integer \(k\) with distribution \(\rho_k\) and \((k-1)\) indices \(i(1), \ldots, i(k-1)\) iid and uniformly random in \([N]\). Then compute a new element of \(\hat{\mathcal{P}}_t\) from the corresponding elements of \(\mathcal{P}_t\): \(\nu = \Psi(\hat{\nu}_{i(1)}, \ldots, \hat{\nu}_{i(k-1)})\).

- The sample \(\hat{\mathcal{P}}_{t-1}\) can be used to estimate the bit error rate after \(t\) iterations. How would you go about it?

Use sampled density evolution to estimate the bit error rate curve \(P_b(t; \varepsilon)\) for \(t \in \{1, 5, 25, 125\}\). Note that for this task, you will have to select \(N\) large enough. You can proceed by trial and error, but \(N \approx 10,000\) is a reasonable starting point.

Also, estimate the asymptotic error curve \(P_b(\varepsilon) = \lim_{t \to \infty} P_b(t; \varepsilon)\) by averaging \(P_b(t; \varepsilon)\) over \(500 < t \leq 1000\).

I expect to receive

1. An explanation of how you estimated the bit error rate from samples.
2. A print-out of the code used for simulations.
3. Plots of the error probability curves predicted by density evolution, comparing them with the empirical results obtained last week.