1 Irregular graph ensembles

One of the most important post-Gallager ideas in LDPC codes was the use of irregular graphs i.e. of graphs with nodes of varying degrees.

The most fundamental irregular graph ensemble can be characterized by the following parameters

1. The blocklength $n$ (ie the number of transmitted bits).
2. The design rate $r$ or, equivalently, the number of check nodes $m = n(1 - r)$ (I am changing notation for the rate).
3. Two distributions $L = \{L_i\}$ and $R = \{R_i\}$ over integers $i = 1, 2, \ldots, d_{\text{max}}$. These are normalized vectors of non-negative numbers:
   \[ \sum_{i=1}^{d_{\text{max}}} L_i = 1 \]  
   \[ \sum_{i=1}^{d_{\text{max}}} R_i = 1. \]
   The probability $L_i$ corresponds to the fraction of variable nodes of degree $i$. Analogously, the probability $R_i$ corresponds to the fraction of parity check nodes of degree $i$.

An ensemble in this class is therefore denoted by $(n, L, R)$. Notice indeed that the number of parity check nodes (and therefore the rate $r$) is uniquely determined by the condition:

\[ n \sum_{i=1}^{d_{\text{max}}} L_i i = m \sum_{i=1}^{d_{\text{max}}} R_i i. \]  

(1)

A graph from the ensemble $(n, L, R)$ is generated using the configuration model: We draw $nL_i$ variable nodes with $i$ half-edges, for each $i \in \{2, \ldots, d_{\text{max}}\}$ and $rR_i$ variable nodes with $i$ half-edges, for each $i \in \{2, \ldots, d_{\text{max}}\}$. We then match the two sets of half-edges according to a uniformly random permutation over $N = n \sum_{i=1}^{d_{\text{max}}} L_i i$ objects.

It is customary (and convenient) to encode the two distributions as generating polynomials:

\[ L(x) = \sum_{i=0}^{d_{\text{max}}} L_i x^i, \quad R(x) = \sum_{i=0}^{d_{\text{max}}} R_i x^i. \]  

(2)

Note that interesting quantities can be easily expressed in terms of these polynomials. For instance, the average variable node degree is $L'(1)$ and the average check node degree is $R'(1)$. The rate is given by

\[ r = 1 - \frac{L'(1)}{R'(1)}. \]  

(3)

Also, the number of edges in the graph is $N = L'(1)n = R'(1)m$.

Another equivalent way of describing the graph consists in assigning the edge-perspective degree distributions

\[ \lambda(x) = \frac{L'(x)}{L'(1)}, \quad \rho(x) = \frac{R'(x)}{R'(1)}. \]  

(4)

These are polynomials $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$, where

\[ \lambda_i = \frac{iL_i}{\sum_j jL_j}, \quad \rho_i = \frac{iR_i}{\sum_j jR_j}. \]  

(5)
Node that \((\lambda_i)_{2 \leq i \leq d_{\text{max}}} \) and \((\rho_i)_{2 \leq i \leq d_{\text{max}}} \) are probability distributions (normalized non-negative vectors). They are sometimes called the size-biasing of \(L, R\). Note that the node perspective degree distributions can be recovered from the edge perspective ones, via

\[
L(x) = \frac{\int_0^x \lambda(y)dy}{\int_0^1 \lambda(y)dy}, \quad R(x) = \frac{\int_0^x \rho(y)dy}{\int_0^1 \rho(y)dy}. \tag{6}
\]

Imagine now you draw a uniformly random edge in the graph \(G\) (out of the \(N\) possible edges). What is the probability that the adjacent variable node has degree \(i\)? There are \(nL_ii\) edges adjacent to a variable node of degree \(i\), hence this probability is

\[
\frac{nL_ii}{N} = \frac{nL_ii}{nL'(1)} = \lambda_i. \tag{7}
\]

Analogously \(\rho_i\) is the probability that, when you draw a uniformly random edge, the adjacent check node has degree \(i\). This is why they are called ‘edge perspective degree distributions.’

Consider next a slightly different experiment. Draw a uniformly random variable node, and look at its neighboring variable nodes. A calculation similar to the one given above shows that they will be distinct and their degrees will be asymptotically i.i.d. with distribution \(\rho\). The neighbors of this check nodes are themselves asymptotically distinct and their degrees i.i.d. with distribution \(\lambda\).

This argument can be extended to any neighborhood \(B_i(t)\) of a random variable node \(i\). This is a random rooted graph whose distribution converges to the one of a random rooted tree which we will call \(T_{\lambda,\rho}(t)\). This is a bipartite tree with offspring distributions \(\lambda\), for variable nodes (except the root), 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 degree distribution of the root is \(L\).

This type of convergence is known in probability theory as ‘local weak convergence’ or ‘Benjamini-Schramm convergence’.

\section{Density evolution}

Using the local weak convergence of the graph \(G\), we can derive the following density evolution recursion

\[
x_{t+1} = \epsilon \lambda(1 - \rho(1 - x_t)). \tag{8}
\]

with initial condition \(x_0 = 1\). In particular, we obtain the following general formula for the threshold of a random code from the \((n, L, R)\) ensemble:

\[
\epsilon_*(\lambda, \rho) = \inf_{x \in (0,1)} \frac{x}{\lambda(1 - \rho(1 - x))}. \tag{9}
\]

The density evolution recursion can also be written as \(\lambda^{-1}(x_{t+1}/\epsilon) = 1 - \rho(1 - x_t)\). Notice that we have

\[
\int_0^1 \epsilon \lambda(x) \, dx = \frac{\epsilon}{L'(1)}, \quad \int_0^1 [1 - \rho(1 - x)] \, dx = 1 - \frac{1}{R'(1)}. \tag{10}
\]

If we represent these relations graphically, they imply that we cannot obtain vanishing bit error rate unless \(\epsilon < L'(1)/R'(1) = 1 - r\). While we knew this already from Shannon channel coding theorem, we obtained an independent proof. Also, this proof gives a crucial insight into how we should design the degree distributions if we want to achieve capacity. Namely we should match the two curves so that \(\lambda^{-1}(x/\epsilon) \approx 1 - \rho(1 - x)\) for all \(x \in (0,1)\).
3 A capacity achieving degree sequence

It turns out that for any given degree distributions $\lambda$, $\rho$ with bounded support the threshold erasure probability (to be denoted as $\epsilon_*(\lambda, \rho)$) is strictly less than the information theoretic bound $1 - r$. We will thus resort to a sequence of degree distributions, to be denoted by $\{\lambda(k), \rho(k)\}$. The design rate of a degree distribution pair will be denoted as $r(\lambda, \rho)$.

Let $\rho^{(k)}(z) = z^{k-1}$, $\hat{\lambda}^{(k)}(z) = \frac{1}{z}[1 - (1 - z)^{1/(k-1)}]$. It follows from the Taylor expansion of $(1 + x)^\alpha$ that

$$
\hat{\lambda}_l^{(k)} = \frac{(-1)^l}{\pi} \Gamma\left(\frac{1}{k-1} + 1\right) \cdot \frac{\Gamma\left(\frac{1}{k-1} - l + 2\right)}{\Gamma(l)}
= \frac{1}{\pi} \frac{\Gamma(l - (k - 1)\cdot(1 - 1))}{\Gamma(l)} \cdot \frac{\sin(\pi/(k - 1))}{(l - 1 - (k - 1)\cdot(1 - 1))}.
$$

(11)

(12)

Let $z_L = \sum_{l=2}^L \hat{\lambda}_l^{(k)}$ and define $L(k, \pi)$ as the smallest value of $L$ such that $z_L \geq 1$. Finally, set

$$
\lambda_l^{(k)} = \begin{cases} 
\frac{\hat{\lambda}_l^{(k)}}{z_L(k, \pi)} & \text{if } l \leq L(k, \epsilon), \\
0 & \text{otherwise.}
\end{cases}
$$

(13)

We claim that: (i) $\epsilon_*(\lambda(k), \rho(k)) > \pi$. Hence this ensemble achieves reliable communication for any $\epsilon \leq \pi$. (ii) $\lim_{k \to \infty} r(\lambda(k), \rho(k)) = 1 - \pi$. Hence the ensemble is capacity achieving.

In order to prove these claims we proceed as follows:

1. Show that $\pi \lambda^{(k)}(1 - \rho^{(k)}(1 - x)) < x$ for all $x \in (0, 1]$, and, as a consequence $\epsilon_*(\lambda^{(k)}, \rho^{(k)}) > \pi$.

Notice indeed that the coefficients $\lambda_l$ in Eq. (11) are non-negative and hence $\lambda_l^{(k)}(x) \leq \hat{\lambda}_l^{(k)}(x)/z_L(k, \pi)$. Therefore

$$
\pi \lambda^{(k)}(1 - \rho^{(k)}(1 - x)) \leq \frac{1}{z_L(k, \pi)} \pi \hat{\lambda}^{(k)}(1 - \rho^{(k)}(1 - x))
= \frac{1}{z_L(k, \pi)} x < x.
$$

(14)

(15)

2. Show that $L(k, \pi) \to \infty$ and $z_L(k, \pi) \to 1$ as $k \to \infty$.

This follows from the fact that $\max_l \hat{\lambda}_l^{(k)} \to 0$ as $k \to \infty$.

3. Notice that, since $\hat{\lambda}_l^{(k)} \leq C/(\ell(k - 1))$, we have

$$
\lim_{k \to \infty} \sum_{\ell=L(k,\pi)+1}^{\infty} \frac{\hat{\lambda}_l^{(k)}}{\ell} = 0.
$$

(16)

and therefore that

$$
\lim_{k \to \infty} r(\lambda^{(k)}, \rho^{(k)}) = 1 - \left\{ \lim_{k \to \infty} \sum_{\ell=1}^{L(k, \pi)} \frac{\lambda_l^{(k)}}{\ell} \right\}^{-1}
= 1 - \left\{ \lim_{k \to \infty} k z_L(k, \pi) \sum_{\ell=1}^{\infty} \frac{\hat{\lambda}_l^{(k)}}{\ell} \right\}^{-1}
= 1 - \pi \lim_{k \to \infty} z_L(k, \pi) = 1 - \pi.
$$
4 Error floor and outer coding

Capacity achieving LDPC codes often present a bad ‘error floor’ behavior. Namely, the bit error probability below threshold vanishes very slowly. Indeed one often gets:

\[ P_b(n, \epsilon) = \frac{K(\epsilon)}{n} + o(1/n), \]  

(17)

The block error rate stays bounded away from 0 as \( n \to \infty \).

Outer coding is a standard approach to address this problem. Before transmitting through the channel, we encode the information message with a code that has high rate \( r_0 \), but very good behavior at low noise (for instance, a code with good minimum distance). This code is not required to be capacity achieving, can be itself a regular LDPC code, or an algebraic code.

We then encode the resulting message using a capacity achieving LDPC code with rate \( r_1 \). Notice that the two blocklength do not need to match, as we can put together several block of the outer code before encoding with the inner code.

At decoding, we first decode the inner code and then the outer code. The total rate is \( r = r_0 r_1 \).

One can think of the inner LDPC code as turning the channel \( \text{BEC}(\epsilon) \) into a channel with very small erasure probability (in fact, with erasure probability of order \( 1/n \) if Eq. (17) is correct).

5 Rateless codes

Given an information sequence \( z \in \{0,1\}^k \), a rateless code produces an infinite sequence of bits \( x = (x_i)_{i \geq 1} \). These are transmitted through a noisy channel which, as we have done so far, we will assume to be a \( \text{BEC}(\epsilon) \). We would like to be able to reconstruct the original information sequence as soon as we collect \( n = k(1 + \delta) \) of the transmitted symbols, for some small \( \delta \). The parameter \( \delta \) is the code overhead.

This can be achieved using a type of ‘low density generating matrix’ (LDGM) codes called LT codes (for Linear Transform or Luby Transform). For each \( i \geq 1 \), we sample an integer with distribution \( (\Omega_k)_{k \geq 1} \) and then transmit the modulo two sum of \( k \) information bits chosen uniformly at random. We will denote by \( \Omega(x) = \sum_k \Omega_k x^k \) the corresponding generating function and define \( \gamma = \Omega'(1)(1 + \delta) \), and \( \omega(x) = \Omega'(x)/\Omega'(1) \).

This defines a sparse random graph that we can use as a basis for a message passing decoder. Density evolution reads

\[
\hat{z}_t = 1 - \omega(1 - z_t),
\]

(18)

\[
z_{t+1} = e^{-\gamma + \gamma z_t}.
\]

(19)

Summarizing

\[
z_{t+1} = \exp \left\{ - (1 + \delta) \Omega'(1 - z_t) \right\}.
\]

(20)

Notice that the bit error rate is bounded away from zero if \( \Omega'(1) < \infty \), and hence in order to achieve vanishing bit error rate we need the average degree do diverge. Matching the two sides of the density evolution equation at \( \delta = 0 \), yields \( \Omega'(x) = -\log(1 - x) \), and hence

\[
\Omega(x) = \sum_{\ell=2}^{\infty} \frac{x^\ell}{\ell (\ell - 1)},
\]

(21)

In other words, for each transmitted, we draw an integer \( \ell \geq 1 \) with probability \( \Omega_\ell = 1/(\ell (\ell - 1)) \), draw \( \ell \) uniformly random information bits and transmit their modulo two sum.

In practice this type of construction suffers from problems at finite blocklength, and hence it is modified by truncating to a large \( \ell_{\text{max}} \), and rescaling it for \( \ell \leq \ell_{\text{max}} \). Also, the residual bit error rate can be improved by outer coding (an approach known as ‘raptor codes’).
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 to optimize irregular ensembles over the erasure channel, and construct capacity-achieving sequences.
3. What is a rateless code, and how the LT construction relates to LDGM codes.

This material can be found in Section 3.15 of MCT, page 109 onwards.

Homework

Write a program that generates a random graph from the \((\lambda, \rho)\) ensemble. Use it to do simulations with the ensemble \((\lambda^{(k)}, \rho^{(k)})\) defined above for \(k = 4, 6, 8, 10\). More precisely: (i) Choose a value of \(\varepsilon\); (ii) Compute the corresponding degree distribution \((\lambda^{(k)}, \rho^{(k)})\); (iii) Generate a random graph with the prescribed degree distribution and blocklength \(n\) of your choice; (iv) Compute bit error probability curves for transmission over BEC(\(\varepsilon\)).

I expect to receive

1. A print-out of the code.
2. The values of the parameters \((\varepsilon, n)\) you used.
3. A plot of the bit error probability curves.