DEPARTMENT OF EE - IIT MADRAS

Assignment 1
Optimizing Irregular LDPC Codes

Surajkumar Harikumar (EE11B075)

November 30, 2014

1 Problem Statement

Find the optimal LDPC code with the highest rate for a given channel erasure probability, and given the maximum left and right degree distributions. To simplify computation, we assume that the check-node degree distribution is a monomial, i.e all the check nodes have the same degree. We assume that the channel is a BEC, and that Gallager's decoding algorithm is used.

2 Solution Scheme

We know that for a (λ, ρ) irregular-LDPC code, the density evolution equation is given by

$$x^{(l+1)} = f(x^{(l)}, \epsilon) = \epsilon\lambda(1-\rho(1-x^{(l)})) \tag{2.1}$$

where $x^{(l)}$ is the probability of edge-error averaged over all edges, in iteration l.

Assuming $\rho(x) = x^{r-1}$, i.e all check nodes are of degree r, we can reduce the equation to

$$x^{(l+1)} = \epsilon\lambda\left(1-(1-x^{(l)})^{r-1}\right) \tag{2.2}$$

We also know that for the probability of error to decay to zero as $l \to \infty$, we require $x^{(l+1)} < x^{(l)}$. This needs to hold for all values of $x \in [0, \epsilon]$. So, we set up a linear programming problem, where we maximize the rate.

$$R = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{1/r}{\sum_0^{l-1} \lambda_i/i} \tag{2.3}$$

So, to maximize rate, we want to MAX $\sum_0^{l-1} \lambda_i/i$

To obtain the inequality constraints, we just vary x in discrete steps from $[0,\epsilon]$. Each density evolution equation corresponds to $f(x,\epsilon) < \epsilon$, so all these equations give us inequality constraints.

We also add the constraint that the sum of all λ_i should be one. $\sum_0^{l-1} \lambda_i = 1$, and that all λ_i lie in $[0,1]$. So our linear programming problem is

$$\begin{aligned} \text{MAX} \sum_0^{l-1} \lambda_i / i & \\ \epsilon \sum_{i=0}^{l-1} \lambda_i \left(1-(1-x^{(l)})^{r-1}\right)^{i-1} &< x \quad \forall x \in \chi \\ \sum_0^{l-1} \lambda_i &= 1 \\ 0 \le \lambda_i &\le 1 \quad \forall i \end{aligned} \tag{2.4}$$

We set up this set of equations, and used MATLAB'S LINPROG function to solve for the best $\lambda(x)$.

Note that we assumed that the check node degree is a constant. We iterate over all values of $r \in [3, r_{max}]$, $r \in \mathbb{N}$ to obtain the best possible LDPC code (assuming constant right degree distribution).

3 Simulation Results and best codes

The function which implements the optimization is **irreg_opt_single.m**. The code for finding the threshold for a given degree distribution is **thresh_finder_bec.m**. A sample code to run the example shown here is in **test1.m**.

Using $\epsilon = 0.48$, we see that $R \le 1 - \epsilon = 0.52$. We set $l_{max} = 100$ and $r_{max} = 12$ and run the optimization procedure. We obtain

$$\begin{aligned} R_{best} &= 0.5187 \\ \rho(x) &= x^{11} \\ \lambda(x) &= 0.2315x^2 + 0.1028x^3 + 0.0667x^4 + 0.0351x^5 + 0.0686x^6 \\ &+ 0.0003x^8 + 0.0586x^9 + 0.0474x^{10} 0.0001x^{15} + 0.0002x^{16} \\ &+ 0.0606x^{17} + 0.0519x^{18} + 0.0002x^{19} + 0.0001x^{35} + 0.0003x^{36} \\ &+ 0.1239x^{37} + 0.0001x^{39} + 0.1150x^{99} + 0.0364x^{100} \end{aligned} \tag{3.1}$$

The best rate-code obtained is **0.5187**, which is very close to the bound of 0.52. We observe that we require nodes of high degree (even 101) to obtain such a high rate LDPC code. One other observation is that for $r \ge 13$, the linear programming problem does not converge. So, this is the best solution scheme for $\rho(x)$ being a monomial, with max degree ≤ 12