Lecture 20

Note Title 3/26/2012

* Irregular LDPC code ensemble

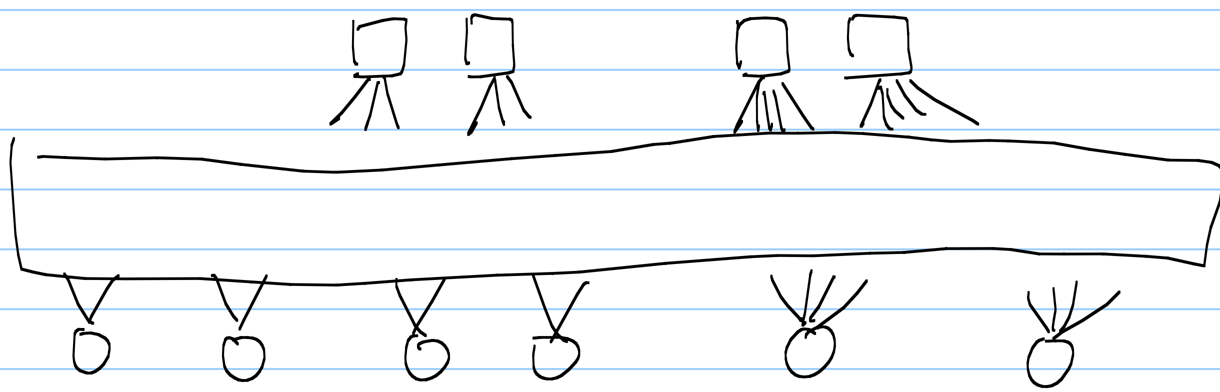
Example: $\frac{2}{3}n$ variables of degree 2

$\frac{1}{3}n$ variables of degree 4

$\frac{1}{3}n$ check nodes of degree 3

$\frac{1}{3}n$ check nodes of degree 5

* Consistency condition; Code rate



* Irregular LDPC code ensemble

Example: $\frac{2}{3}n$ variables of degree 2

$\frac{1}{3}n$ variables of degree 4

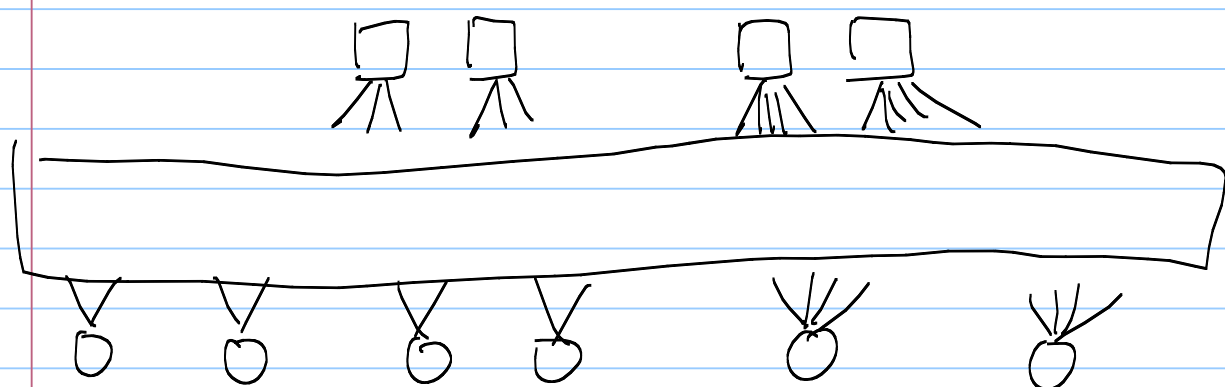
$\frac{1}{3}n$ check nodes of degree 3

$\frac{1}{3}n$ check nodes of degree 5

* Consistency condition

Total # edges: $\frac{2}{3}n \times 2 + \frac{1}{3}n \times 4 = \frac{1}{3}n \times 3 + \frac{1}{3}n \times 5$

When $n = 6$, we have the following FG.



* It is critical to keep the edge count consistent when we design the sockets of the var. & chk. nodes separately.

* It is a rate $\frac{1}{3}$ code

(n var nodes, $\frac{2}{3}n$ check nodes)

Q: How to perform the DE analysis?

Variable

$\deg = 2$ $\qquad$ $\deg = 4$

$Q_{c \to v}$ $\qquad$ $Q_{c \to v}$

Observation 1: again, all incoming messages are i.i.d. due to the random interleaver

Check

$\deg = 3$ $\qquad$ $\deg\ 5$

$P_{v \to c}$ $\qquad$ $P_{v \to c}$

Observation 2: assume we have the incoming message distribution $P_{v \to c}$ & $Q_{c \to v}$, we can compute

Q: How to compute the update rule?

$$P^{(t)}_{v \to c} = \lambda_2 P^{\deg=2}_{v \to c} + \lambda_4 P^{\deg=4}_{v \to c}$$

Further average

$$= \lambda_2 \left(F_{var}\left(P^{(0)}, 2, Q^{(t-1)}_{c \to v}\right) \right) + \lambda_4 \left(F_{var}\left(P^{(0)}, 4, Q^{(t-1)}_{c \to v}\right) \right)$$

$$Q^{(t)}_{c \to v} = \rho_3 \left(F_{chk}\left(P^{(t)}_{v \to c}, 3\right) \right) + \rho_5 \left(F_{chk}\left(P^{(t)}_{v \to c}, 5\right) \right)$$

$P^{\deg 2}_{v \to c}$

$P^{\deg 4}_{v \to c}$

$Q^{\deg 3}_{c \to v}$

$Q^{\deg 5}_{c \to v}$

Q: What are the suitable values of

$\lambda_2, \lambda_4, \rho_3, \rho_5$?

Ans: $\lambda_2 = \dfrac{\frac{2}{3}n \times 2}{\frac{8}{3}n} = \dfrac{1}{2}$

edges emiting from a deg 2 node

total # of edges

$\lambda_4 = \dfrac{1}{2}$

$\rho_3 = \dfrac{\frac{1}{3}n \times 3}{\frac{8}{3}n} = \dfrac{3}{8}$

edges emitting from a deg 3 ; check node

total # of edges

$\rho_5 = \dfrac{5}{8}$

We call λ_2, λ_4, the "edge-based" var degree distribution

ρ_3, ρ_5 the edge-based chk deg distribution

For general edge-based deg distributions $\lambda_2, \cdots, \lambda_{\max d_v}$

$\rho_2, \cdots, \rho_{\max d_c}$

$$Q^{(t)} = \sum_{d=2}^{\max d_c} \rho_d F_{chk}(p^{(t)}, d-1)$$

$$p^{(t)} = \sum_{d=2}^{\max d_v} \lambda_d F_{var}(p^{(0)}, Q^{(t-1)}, d-1)$$

In the coding community, the deg distribution is usually described by a polynomial

$$\lambda(x) = \sum_{k=2}^{\max d_v} \lambda_k x^{k-1}$$

$$\rho(x) = \sum_{k=2}^{\max d_c} \rho_k x^{k-1}$$

and we say a (λ, ρ) irregular code ensemble

We require $\lambda(1) = \rho(1) = 1$

Exercise:
given $\lambda(x) = 0.5x^2 + 0.5x^3$

$\rho(x) = 0.5x^4 + 0.5x^5$

Construct one such irr. code

Exercise:
The code rate of an irregular (λ, ρ) code is

$$1 - \frac{\int_0^1 \rho(s)\,ds}{\int_0^1 \lambda(s)\,ds}$$

This special form of deg polynomials is very convenient.

Ex: $\lambda(x) = 0.5x^2 + 0.5x^3 \quad \rho(x) = 0.5x^4 + 0.5x^5$

Write down the DE formulas for BECs.

Ans: Suppose we are facing a BEC w. erasure ε.

$$p^{(0)} = \varepsilon, \qquad p^{(1)} = p^{(0)}$$

$$q^{(t)} = \sum_{k=2}^{\max d_c} \rho_k \left(1 - (1-p^{(t)})^{k-1}\right)$$

$$= 1 - \rho(1 - p^{(t)})$$

$$p^{(t+1)} = \sum_{k=2}^{\max d_v} \lambda_k \left(p^{(0)} \left(q^{(t)}\right)^{k-1}\right)$$

$$= p^{(0)} \lambda(q^{(t)})$$

Sometimes the overall density evolution can be summarized as

$$\boxed{p^{(t+1)} = \varepsilon \cdot \lambda\left(1 - \rho(1 - p^{(t)})\right)}$$

* DE analysis thus applies to irregular LDPC codes as well

* How to use DE to design your own good LDPC codes of a given code rate R?

① Fix max d_v max d_c. Randomly choose

$\lambda_2, \lambda_3, \cdots, \lambda_{\max d_v}$ that satisfies the rate constraint

$\rho_2, \rho_3, \cdots, \rho_{\max d_c}$ $\quad 1 - \dfrac{\int_0^1 \rho\, dx}{\int_0^1 \lambda\, dx} = R$

② Suppose the channel is parametrized by σ, Run DE for to say $t_0 = 100$, to see whether $p_{ber}^{(t)} < 0.001$ (means convergence to zero)

③ Run a binary search to find σ^* of the given λ, ρ

$\boxed{\text{Hill-climbing algorithm, or any other general optimization techniques}}$

④ Slightly perturb λ, ρ while maintaining the code rate R, let λ^{new}, ρ^{new} denote the perturbed version.

find the $\sigma^{*,new}$ for λ^{new}, ρ^{new}

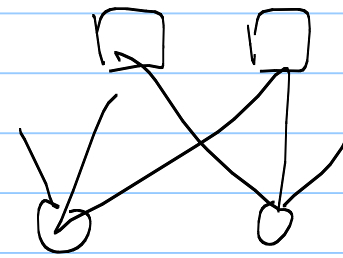
⑤ Retain the better degree distribution pair. Repeat ④

(5.1) Sometimes we may be trapped in a local optimum point. Choose a different starting point & repeat ① — ⑤

⑥ With the optimized λ^*, ρ^*

construct a code with the desired codeword length. See the previous exercise.

Sometimes we have to quantify the λ^*, ρ^* to fit the constraint of using a finite n

7. Arbitrarily select an interleaver with no obvious short cycles ex: cycle of length 2  length 4 

8. Convert the graph to the H matrix & G matrix for encoding

Voila! Very good performance LDPC codes.

DE holds for 1. any cycle-free FG, or any asymptotically-cycle-free networks.

2. Symmetry condition $\Rightarrow$ all-zero codeword

3. identically distributed messages

   $\Rightarrow$ remove the dependence on the subscript.

Pros: DE is thus quite general & predicts the performance well. (Think general large FG, rather than error control codes)

Cons: ① The symmetry condition ② is too strong.

Ex: Z channel for optical storage.

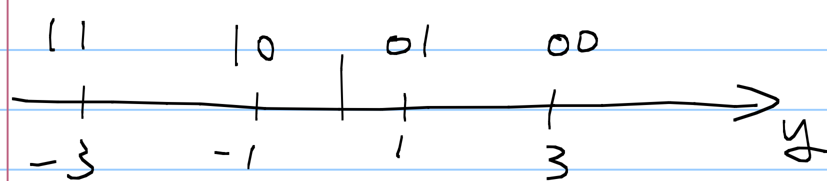
Example:

$$Y = 2(-1)^{X_1} + (-1)^{X_2} + N$$ in the Euclidean space.

X_1, X_2 are Bernoulli with $p = 0.5$

N is standard Gsn.

the constellation points of this PAM are

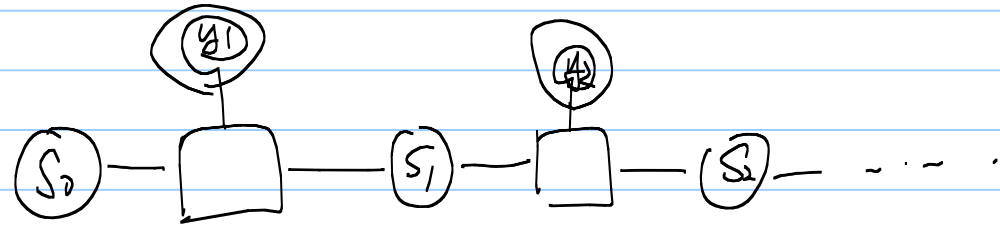


Note: This system is not symmetric, as the codewords 00, & 11 have smaller prob being misdected as other codewords

$$P(\text{error} \mid \vec{X} = 00) < P(\text{error} \mid \vec{X} = 01)$$

② The <u>density evolution</u> computation is too complicated, since we are essentially computing the density of a random function when focusing on a general FG.

Ex: we have a linear convolutional code,



Use the DE to analyze the performance

Ans: Theoretically doable as it is a cycle-free network.

However, each message α or β is a $|S|$-dimensional vector.

We are tracing the density of $|S|$-dim vectors. $\Longrightarrow$ Hard.

On the other hand, analyzing convolutional codes is an important question as it relates to the turbo code performance

For the following, we will study more the DE & try to answer these challenges at least partially.

* Deal with the 2nd drawback of DE (at least for general FGs), the complexity

Note: for binary LDPC codes, the exact DE is fast

* Instead of tracing the exact density, we trace the approximate density.

We sacrifice some precision of the prediction for better efficiency.

* Gaussian Approximation (Too many variants of Gsn Approx. We discuss some general procedures.)

Motivation 1: For AWGN channels: $Y=(-1)^X+\sigma N$

the initial message distribution is

$$P_{m_i|\vec{X}=\vec{0}} \sim \text{Gsn}\left(\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\right) \quad \text{Gsn.}$$

Motivation 2: If all incoming messages are Gsn, then the variable node message map

$$m^{(t+1)}_{X_i C_j} = m_i^{(0)} + \sum_{l \in \partial i \setminus j} m^{(t)}_{C_l X_i}$$

preserves the Gaussianality.

Remark: Unfortunately, the check node map

$$m^{(t)}_{c_j x_i} = 2\tanh^{-1}\left(\prod_{k\in\partial j\setminus i} \tanh\left(\frac{m^{(t)}_{x_k c_j}}{2}\right)\right)$$

does not preserve the Gaussianality.

Motivation 3: If d_v is large,

$$m^{(t+1)}_{x_i c_j} = m^{(0)}_i + \underbrace{\sum_{l\in\partial i\setminus j} m^{(t)}_{c_l x_i}}_{d_v - 1 \text{ terms}}$$

then the var. message map will "restore" the Gaussianality due to the central limit theorem.

Remark: ① CLT requires normalization
without normalization, simply taking convolution does not give you Gsn

② CLT requires medium-sized d_v.
In practice, d_v is very small,
(mostly 2 or 3, generally ≤ 6)