Lecture 19

Note Title 3/21/2012

* Density evolution & its related concepts will be our main subjects for 2 weeks.

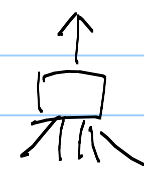
* Cycle-free, asymptotic analysis

* Use the marginals of the incoming messages to compute the marginal of the outgoing messages through a <u>message map</u>.

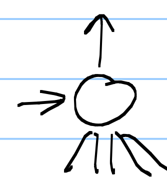
* For i.i.d noise, we can drop the subscript of the DE

$$P^{(1)} = P^{[0]}$$

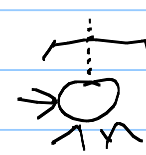
$$Q^{(t)} = F_{chk}\left(P^{(t)}, d_c - 1\right)$$



$$P^{(t+1)} = F_{var}\left(P^{(0)}, Q^{(t)}, d_v - 1\right)$$



$$P_{out}^{(t_0)} = F_{final}\left(P^{(0)}, Q^{(t_0)}, d_v\right)$$

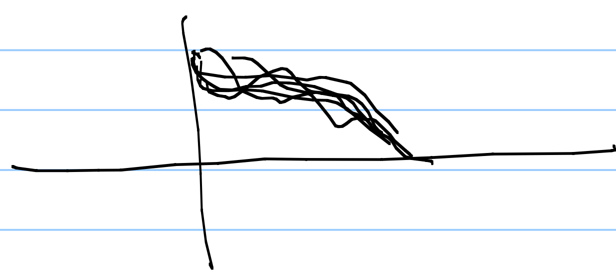


$$P_{b,DE}^{[t_0]} \triangleq P\left(m^{(t_0)} < 0 \mid X = 0\right) + \frac{1}{2} P\left(m^{(t_0)} = 0 \mid X = 0\right)$$

* For a code C chosen from the (n, d_v, d_c) ensemble. We let $P^{(t_0)}_{C,\,ber,\,x_i}$ denote the actual ber performance of x_i when running the belief-propagation for t_0 times. For any ε, t_0, we have

$$\lim_{n\to\infty} P_C\left(\left|\left(\frac{1}{n}\sum_{i=1}^{n} P^{(t_0)}_{C,\,ber,\,x_i}\right) - P^{(t_0)}_{ber,\,DE}\right| > \varepsilon\right) = 0$$

Namely, when $n\to\infty$
with high prob that the performance of a randomly constructed code concentrates around the avg of the ensemble predicted by DE.



Pf: Not very surprising.

From the bit-perspective, each bit has a cycle-free support tree with ≈ 1 prob.

$\Rightarrow$ Most bits have a cycle-free support tree. & have performance predicted by $P^{(t)}_{ber,\,DE}$

$\Rightarrow$ The performance of the "bad bits" is insignificant when averaging over all bits.

Comparison of DE with Monte-Carlo simulation

MC: ① randomly corrupt the codeword ② decode & avg the perf.

DE	MC
∞ codeword length	finite codeword length n
Deterministic evaluation Important for code optimization when comparing two codes (4, 8) vs. (3, 6)	Results will vary
Complexity: Linear with respect to to	linear to to, n, & the number of trials $\propto \left(\frac{1}{\text{desired BER}} \right)$
capable of extremely small ber	ber $10^{-5} \sim 10^{-8}$

especially when the code performances are close to each other

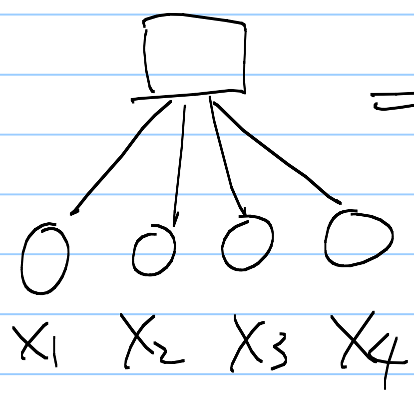
* How to compute numerically $P^{(t)}$ $Q^{(t)}$ for other channel distributions? (How to implement the DE?)

Ans: ① Quantification → exact evolution by exhaustive enumeration of the joint prob & the redistribution of the weight after the mappings

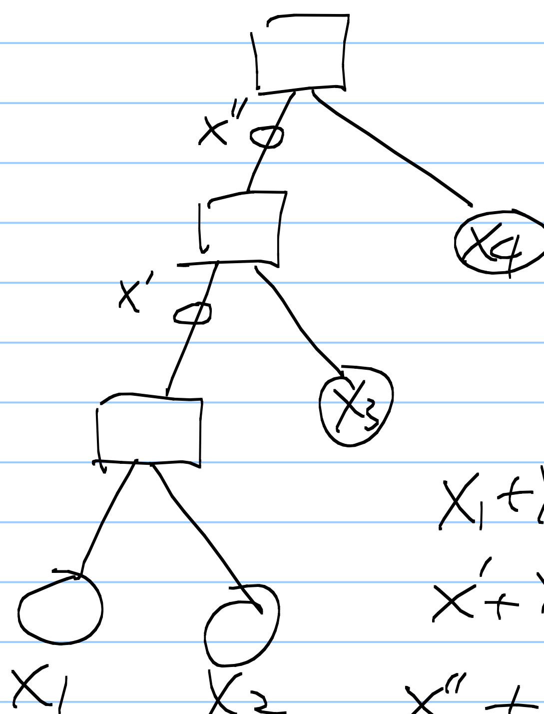
② Pairwis computation

Var: $$P_1 * P_2 * P_3 = (P_1 * P_2) * P_3$$

Chk:



=



$$X_1 + X_2 + X_3 + X_4 = 0$$

$$X_1 + X_2 + X' = 0$$
$$X' + X_3 + X'' = 0$$
$$X'' + X_4 = 0$$

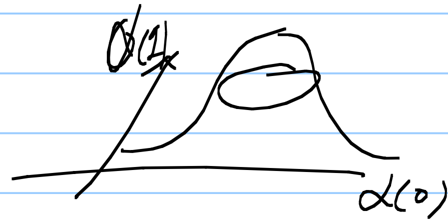
$$\Rightarrow F_{chk,dc}(\quad) = F_{chk2}(F_{chk,12}(F_{chk,d}\ , P^{(t)}), P^{(t)})$$

Generalization

① The DE is applicable to any symmetric message-passing algorithms, such as the VA-algorithms, <u>quantized</u> sum-product algorithm. (the messages take values from a small finite set)

② The DE depends on the form of the messages.

For example, if we use $\alpha(\cdot)$ $\beta(\cdot)$ as the messages, each function $\alpha(\cdot)$ can be viewed as a 2-dim vector $(\alpha(0), \alpha(1))$. Then the density to trace is



2-dim density

Error prob becomes

$$P(\alpha_{out}(1) > \alpha_{out}(0) \mid X_i = 0) + \frac{1}{2} P(\alpha_{out}(1) = \alpha_{in}(0) \mid X_i = 0)$$

* Irregular LDPC code ensemble

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

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

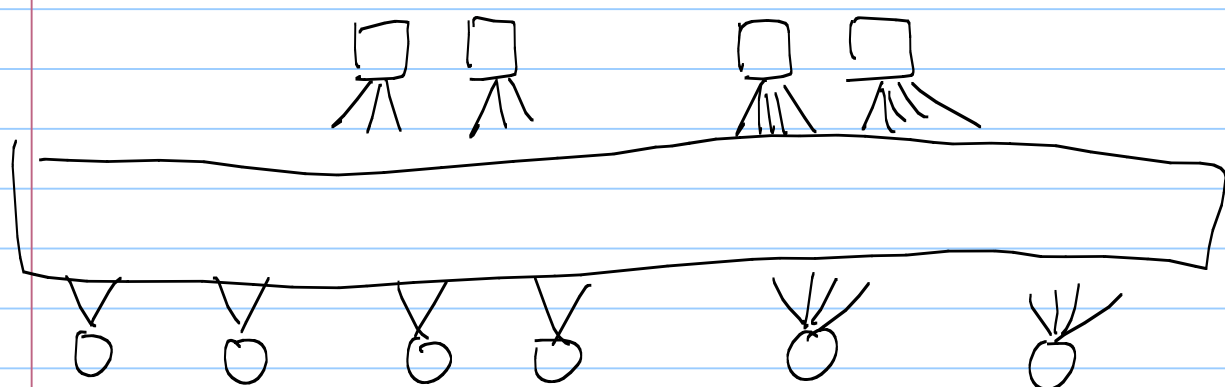
$\frac{1}{3}n$ check nodes of degree 3

$\frac{1}{3}n$ 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.

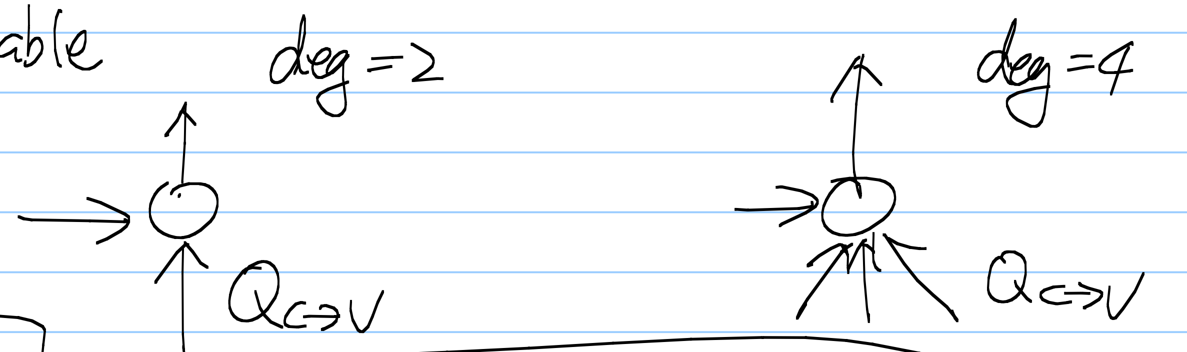


* It is critical to keep the edge count consistent when we design the sockets of the var. & chk. nodes separately.

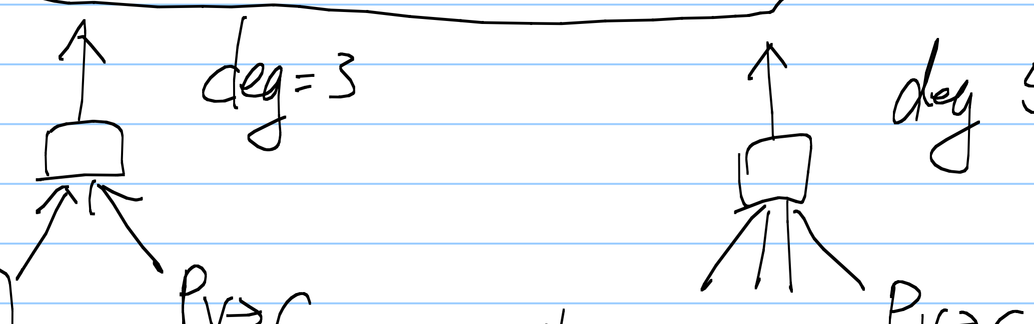
* It is a rate $\frac{1}{3}$ code

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

Q: How to perform the DE analysis?



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



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

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

$P^{deg\ 2}_{v\to c}$
$P^{deg\ 4}_{v\to c}$
$Q^{deg\ 3}_{c\to v}$
$Q^{deg\ 5}_{c\to v}$

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