

Lecture 36: LDPC Codes Round-up

November 19, 2007

Lecturer: Atri Rudra

Scribe: Thanh-Nhan Nguyen

We refer readers to lecture 34 for Iterative Message Passing Decoder. In this lecture, we will cover the peeling algorithm, irregular LDPC codes, and expander codes.

1 Peeling Decoding

1.1 Observations

Before describing the peeling algorithm, we have following observations:

- (i) Do not need to send an erasure explicitly (no ? message)
- (ii) Once c_i knows its value, it can "leave"

1.2 Peeling Algorithm

Step 1: Every check node p_j (and a neighbor c_i) maintains the XOR of all message bits from all its neighbors other than c_i (call it m_{ij})

Step 2: Do message passing as before except:

- (i) If an ? needs to be sent then do not send any message
- (ii) Once c_i knows its value $b_i \in \{0, 1\}$ then remove the variable node and its incident edges after sending the message
- (iii) If a check node p_j receives all correct message bits from all its neighbors other than c_i then delete (c_i, p_j) after sending the message (if all edges incident to p_j are removed then remove p_j). Otherwise, update m'_{ij} s

We have followings claims

Claim 1: The peeling algorithm correctly implements the message passing algorithm.

Proof. We leave this as an exercise. Hint: this can be proved by using induction. □

Claim 2: For every edge in the factor graph, at most one message is sent. Therefore, the running time = $O(\text{the number of edges}) = O(n)$

Proof. An edge is deleted as soon as a message is passed. □

2 Irregular LDPC codes

Regular LDPC codes do not achieve BEC_α capacity. Yet, Luby et. al showed that irregular LDPC codes do! Note that with irregular LDPC codes all nodes on left (or right) do not have the same degree.

We observe that: for variable nodes, having more neighbors is better as chance of getting the correct bit is higher. For check nodes, having less neighbors is better as chance of getting ? is lower. Then, d_v should be larger, and d_c should be smaller. Or, the number of check nodes is greater than the number of variable nodes. In this case, it is impossible to get positive rate $1 - \frac{d_v}{d_c}$. With irregular LDPC codes, we have more freedom to achieve: the number of check nodes \geq the number of variable nodes.

Intuition for better performance: Variable nodes with high degree imply faster convergence to correct bit value. Then, its neighbor check nodes get the correct bit value faster. And, variable nodes with small degree get correct value faster.

Theorem 2.1. *Can get within ε of capacity of BEC_α with encoding and decoding time of $O(n \lg(\frac{1}{\varepsilon}))$*

However, it is not an explicit construction but a randomized construction.

For BSC_p , work well experimentally, but no theoretical guarantees known.

Open questions:

1. Explicit irregular LDPC codes that achieve BEC_α capacity.
2. Prove theoretically that irregular LDPC code achieve BSC_p capacity.

3 Expander Codes

Expander codes are explicit regular LDPC codes. Factor graph is an “expander” graph.

Denote S a subset with $|S| \leq n/2$. Expander property:

$$|r(S)| \geq \Omega(d|S|)$$

where $r(S)$ is the set of all neighbor of S .

3.1 Properties

1. This is a explicit asymptotically good (binary) codes, the only such codes that we are not based on code concatenation
2. Expander codes can correct some $p > 0$ worst-case errors in linear time (message passing algorithm)
3. Encoding is linear time