PolarCode User Manual

example_run.m shows how to use the implemented codes.

Features

Three code construction ：

1. Heuristic Method via Bhattacharyya parameter （channel dependent） [1]
2. Density Evolution Gaussian Approximation （channel dependent） [2]
3. Huawei Approximation (channel independent） [3]

Encoding

1. 3GPP encoding (without bit-reversal permutation) $\mathbf{x} = \mathbf{u}\mathbf{F}_n$
2. Original Arikan proposed (with bit-reversal permutation) $\mathbf{x} = \mathbf{u}\mathbf{B}_n\mathbf{F}_n$

Decoding

1. SC decoder
2. CRC-aided SC decoder

Rate matching and rate recovery

1. to support different codeword length and code rate

Function list

Code Construction:

- *PolarCodeConstrct.m* : figure out frozen bit positions and puncture pattern
- *channel_polarization_BP*.m : implement Heuristic Method via Bhattacharyya parameter
- *channel_polariztion_GA*.m: implement Gaussian Approximation
- *channel_polarization_huawei_approx*.m: implement Huawei Approximation

Encoding:

- *polar_code_encoder.m*

Decoding:

- *polar_code_sc_decoder.m*
- *Polar_code_sscl_decoder.m*

Rate matching and rate recovery

- *polar_code_rate_matching*
- *polar_code_rate_recovery*

Helper functions:

- *compute_crc*.m
- crc_generator.m

Core functions implementation explanation

Channel Polarization:

- Heuristic Method via Bhattacharyya parameter [1] (Eq.6 or Eq. 38]
- Density Evolution Gaussian Approximation [2] Algorithm 1
- Huawei Approximation [3]

Encoding

For 3GPP

$$\begin{aligned}
\mathbf{c} = \mathbf{u}\mathbf{G}_N \\
&= [\mathbf{u}_1, \mathbf{u}_2] \begin{bmatrix} G_{\frac{N}{2}} & 0 \\ G_{\frac{N}{2}} & G_{\frac{N}{2}} \end{bmatrix} \\
&= \left[(\mathbf{u}_1 \oplus \mathbf{u}_2) G_{\frac{N}{2}}, \mathbf{u}_2 G_{\frac{N}{2}}\right] \\
&\mathbf{u} \in \mathbb{B}^{1\times N} \\
&\mathbf{u}_1 = \left[u_1, u_2, \ldots, u_{N/2}\right] \\
&\mathbf{u}_2 = \left[u_{N/2+1}, u_{N/2+2}, \ldots, u_N\right]
\end{aligned}$$

For Arikan original

Adapted from the original derivation from[1](Eq. 68 page 13) and [4](page 4)

$$\begin{aligned}\mathbf{c} &= \mathbf{u}\mathbf{B}_N\mathbf{G}_N \\ &= [\mathbf{u}_1, \mathbf{u}_2]\begin{bmatrix} G_{\frac{N}{2}} & 0 \\ G_{\frac{N}{2}} & G_{\frac{N}{2}} \end{bmatrix} \\ &= \left[(\mathbf{u}_1 \oplus \mathbf{u}_2)\, G_{\frac{N}{2}}, \mathbf{u}_2 G_{\frac{N}{2}}\right] \\ \mathbf{u} &\in \mathbb{B}^{1\times N} \\ \mathbf{u}_1 &= [u_1, u_3, \ldots, u_{N-1}] \\ \mathbf{u}_2 &= [u_2, u_4, \ldots, u_N]\end{aligned}$$

With the above equation, the encoding function($f(\mathbf{u})$) can be implemented recursively as

$$f(\mathbf{u}) = [f(\mathbf{u}_1 \oplus \mathbf{u}_2), f(\mathbf{u}_2)]$$

SC decoding (non-bit reversal permutation):

For each non-leaf code, it receive a soft bits vector from its parents and send the corresponding hard decision vector to its parent. refer to the following diagram. It invoves 6 messages and 3 operations in the following sequence

1. node i receive a soft bits vector from its parent node $\lfloor \alpha_v \rfloor$
2. node i perform the first operation (check node operation) to calculate soft bits vector α_l for its left child: node $2i$

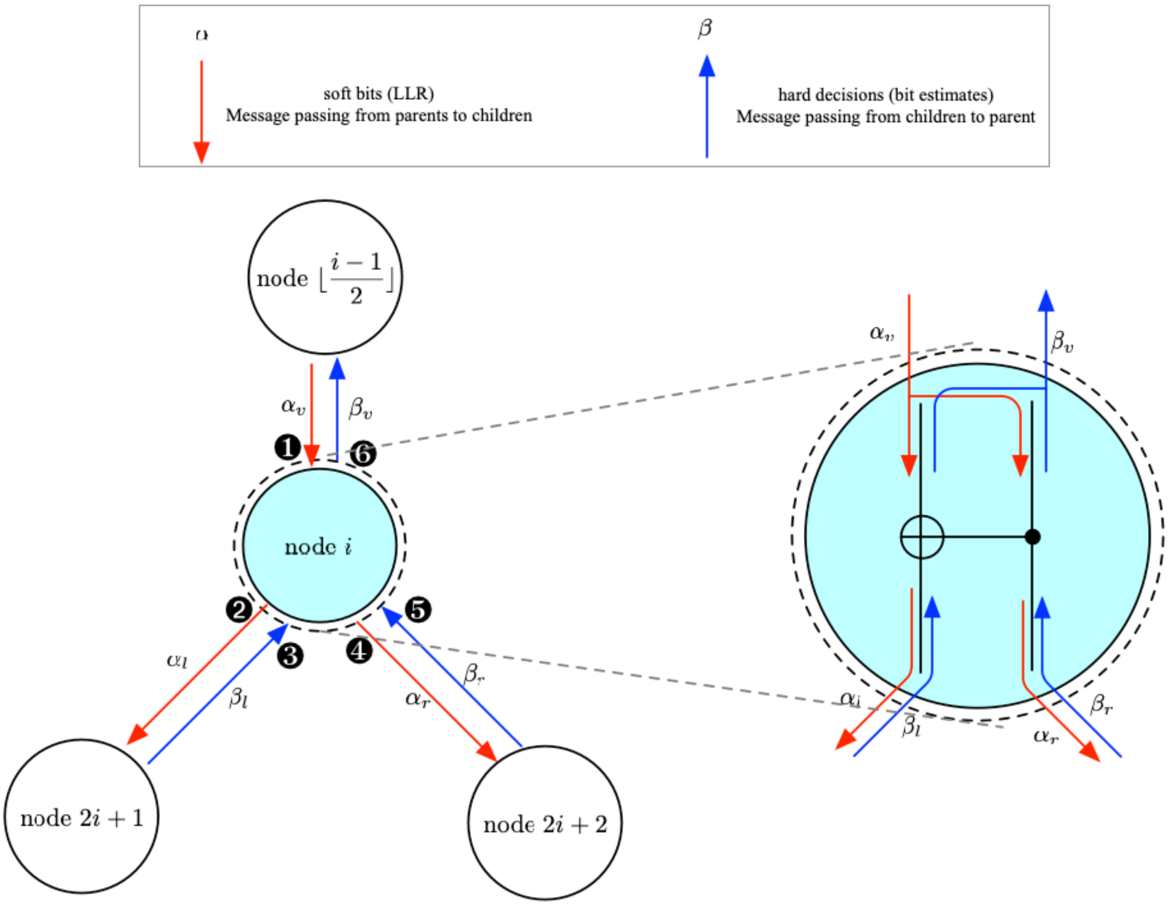
$$\begin{aligned}\alpha_l &= f(\alpha_v^{(1)}, \alpha_v^{(2)}) \\ &= 2\text{atanh}(\tanh(a/2)\tanh(b/2)) \\ &\approx \text{sgn}(a.*b)\min(|a||b|)\end{aligned}$$
$$\alpha_v^{(1)} = [\alpha_{v,1}, \alpha_{v,2}, ..., \alpha_{v,N/2}], \alpha_v^{(2)} = [\alpha_{v,N/2+1}, \alpha_{v,N/2+2}, ..., \alpha_{v,N}], N = \text{card}(\alpha_v)$$

3. node i receives the hard decision β_l from its left child
4. node i perform the 2nd operation (bit node operation) to calculate soft bits α_r for its right child: node $2i+1$ and sent it .

$$\alpha_r = (1 - 2\beta_l)b + a$$

5. node i received the hard decision β_r from its right child
6. node i perform the third operation, combine β_l and β_r to obation β

$$\beta = [\beta_l \oplus \beta_r, \beta_r]$$



For each leaf node:

- if it corresponds to a frozen bit, the hard decision will always be 0
- if it corresponds to a information bit, the hard decision performed according to threshold detection, soft bits > =0, hard decision = 0, soft bits < 0, hard decision = 1

Note: for the bit-reversal permutation case, simply change a and b to the odd and even bits of α_v, β also needs change orders accordingly, refer to the code in *polar_code_sscl_decoder.m.*

SCL decoding

Run L SC decoders in parallel, all operations for non-leaf node are the same. when reaching a leaf node, for information bit instead of make threshold detection, it generate two candidates: the candidate agrees with threshold detection and the candidate oppose threshold detection. A penalty is put on the oppose candidate. For nL decoders, it becomes 2L candidates. Then a sort and prune operation based on path metrics to select nL candidates with least path metrics.

The idea is simple, but in terms of implementation, it has a very high complexity in terms of time and memory. As the information(soft bits, hard decisions, PM) about each node of each SC encoder need to be stored and exchanged and compared when needed. The time complexity is $O(LN^2)$. The memory capacity is $O(LN^2)$

This implementation optimizes the process, in stead of save all information of each node for all SC encoders, each encoder has a shared storage of soft bits which changes(push in and pop out)accordingly to ensure only necessary soft bits staying in the storage. This reduced time/memory complexity to $O(LN\log_2(N))$

Reference

[1] E. Arikan, “Channel polarization: A method for constructing capacityachieving codes for symmetric binary-input memoryless channels,” CoRR, vol. abs/0807.3917, 2008.

[2]H. Ochiai, P. Mitran and H. Vincent Poor, "Capacity-Approaching Polar Codes With Long Codewords and Successive Cancellation Decoding Based on Improved Gaussian Approximation," in IEEE Transactions on Communications, vol. 69, no. 1, pp. 31-43, Jan. 2021, doi: 10.1109/TCOMM.2020.3030057.

[3] 3GPP R1-167209. Polar code design and rate matching

[4] Pfister, Henry D. "A brief introduction to Polar codes." Supplemental Material for Advanced Channel Coding (2014).

[5] Ochiai, Hideki, Patrick Mitran, and H. Vincent Poor. "Capacity-Approaching Polar Codes with Long Codewords and Successive Cancellation Decoding Based on Improved Gaussian Approximation." IEEE Transactions on Communications 69.1 (2020): 31-43.