Lecture 9 LDPC Codes (Low Density Parity Check)

* We saw that Turbo Codes were an evolution of convolutional codes with parallel concatenation and interleaver

* LDPC codes are an "evolution" of block codes

↳ Quick reminder of basic parameters:

k: number of information bits

n: number of code bits

R: code rate k/n

d_{min}: minimum distance

↳ Reminder of Parity check Matrix

$\bar{x} \cdot H^T = 0$, $\bar{x} = \bar{m} G$

↳ Ways to describe the linear code block

Codebook (list of codewords)

Parity-check Matrix / Generator Matrix (Also polynomial)

Tanner Graph (Graphical representation)

↳ Reminder Systematic code:

In the first position we have the original codeword and in the second part we have the parity check control (codeword Parity)

and the generator matrix can be written: $G = [I_k | P]$

Tanner Graphs

* It consists of two sets of vertices

n - vertices for the codeword bits (bit-nodes)

$m = n-k$ vertices for the parity check equations (check nodes)

* Bit node j is connected to check node $i \Leftrightarrow h_{ji}$ of H is equal 1

* The H matrix can be entirely described by Tanner Graph

↳ The check nodes test and if the rules of the code are respected then the check nodes will be equal to zero

LDPC Characteristics

* LDPC codes are Linear Code Blocks with a parity check matrix H^T that contains only a very very few number of "1"s

$H = (H^T)^T$

⇒ Sparseness of H matrix guarantees an increase only linear with the code length of the SOFT decoding complexity

⇒ If a block code has a sparse H ⇒ It's a LDPC code, but it is not easy to find a sparse H for the traditional block codes ⇒ so we cannot implement easily soft decoding

⇒ The difference is:
- ↳ Classic block codes are decoded with Maximal Likehood Hard Decoding
- ↳ LDPC codes are SOFT-Decoded iteratively

* Definitions

•) A LDPC code is called regular is H(m,n) contains:
1. A fixed number $w_c \geq 3$ of "1s" for each column
2. A fixed number w_r of "1s" for each row, where

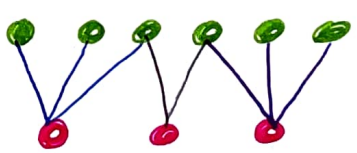
$m \cdot w_r = n w_c \qquad w_c << m = n-k$, k = number of message bits

•) A LDPC is called irregular if H(m,n) does not contain a constant number of "1s" for each column

⇒ We can check in the Tanner graph if the code is regular
- ↳ If every bit node has the same number of links
- ↳ If every check node has the same number of links

} Regular code

Ex: $H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$ message nodes / check nodes



Decoding Strategies of LDPC Codes

~~We can apply the soft decision to a LBC by using iterative decoding algorithms.~~

The algorithms used to decode LDPC codes are Message-Passing algorithms (or iterative algorithms). Depending on messages are called differently

↳ If message is binary ⇒ bit-flipping decoding (hard decision)

↳ If message are probabilities (by likehood) ⇒ belief propagation decoding (soft decisson)

BIT-Flipping Decoding

There are two steps:

① → Bit nodes send the info to the check nodes

② → Check nodes verify Parity Check Equatsons (PCE) and transmit the result to bit nodes

Analysing iteratively the bit nodes and PCE the algorithm inverts the value of probable wrong bits (flipping bits)

At the end of any iteration the it is possible to find all correct values for the codeword

* Example hard decoding

We have:

$$H=\begin{bmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\1&0&1&1&0&0&1\end{bmatrix}$$

we send

$c=1011001$

but the noise:

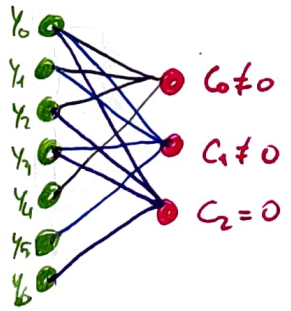
$y=c+[0\,1\,0\,0\,0\,0\,0]$

$y=[1\,\textcircled{1}\,1\,1\,0\,0\,1]$

wrong

Observing the Tanner graph we can see that the C_2 is the only correct:

⇒ We can assume that wrong bits are not linked to node C_2



So, the decoder will check only y_1, y_4, y_5 inverting iteratively (flipping) these bits

SUM-Product Algorithm (Soft Decoding)

We use a belief propagation algorithm

The idea:

* Before the observation we only know the a priori prob. ($P(x_4)$)
* After the observation we have the a posteriori prob. of the bit given the bit that was sent $P(x_4/y_4)$
* Then we try to give the propagation of the information (prob. propag.)
* Finally, we end in the Maximum A Posteriori decision (MAP decision) that says:
  ↳ What is the prob. of the bit given the rest $P(x_4/y_1 y_2 \cdots y_6)$

Performance

The LDPC codes get better performance than turbo codes

* The secret is:
  → Because there are few "1s" the minimum distance is very small (not the key point!)
  To increase d_{min} we have to increase the number of "1s" and therefore the algorithm becomes more complex ⇒ not possible SOFT decoding
  → The point is the length (Shannon said: Long and random codes could achieve capacity limits)

LDPC codes vs TURBO codes

Advantages of LDPC

→ Not necessary to know the noise parameter
→ Possible to stop the iteration process
→ Better Performance
→ Only one decoder (Turbo needs 2)
→ Smaller iteration decoding complexity

Disadvantages

→ LDPC codes need more iterations
→ Construction process of LDPC codes is more complicated than the construction process of a Turbo Code