Lecture 1 : Quantum LDPC Codes

LDPC = low density parity check

This term comes from classical coding theory so that is where we will start!

Recap : Linear Codes

A (classical) linear code C is a subspace of the vector space $\mathbb{F}_2^n$ ie vectors of length n with entries in $\{0,1\}$

and addition carried out mod 2.

We can specify C via its **parity-check matrix** $H \in M_{m \times n}(\mathbb{F}_2)$

ie H is an m by n matrix with entries in $\mathbb{F}_2$.

We have $\boxed{C = \ker H}$

where $\ker H = \{ v \in \mathbb{F}_2^n \text{ s.t. } Hv = 0 \}$.

We interpret 0 as the zero vector here. In words, C contains all vectors that have even overlap with all the rows of H.

We call these vectors **codewords**.

Example: $H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ "means generated by"

$\ker H = \langle (0,0,0)^T, (1,1,1)^T \rangle$

We note that 0 is always in $\ker H$ for any H so 0 is always a codeword of any linear code.

Our example is simply the **repetition code**!

A linear code has 3 important parameters:

- n number of (physical) bits

- k number of (encoded) bits
  also called the code dimension

- d the code distance

For $H \in M_{m \times n}(\mathbb{F}_2)$ we have

$$\boxed{k = n - \operatorname{rank} H}$$

where we recall that the rank of a matrix is equal to the number of linearly independent rows (or columns) in the matrix.

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

$\operatorname{rank} H = 2$ so $k = 3 - 2 = 1$

To compute the rank of a matrix we apply Gaussian elimination & so finding the dimension of a linear code is *efficient*.

To define the code distance, we first recall the defn of the (*Hamming) weight* of a binary vector $v \in \mathbb{F}_2^n$.

$wt(v) = \#$ of nonzero entries in v.

We can then define the code

distance of a liner code

C to be

$$\boxed{d = \min_{v \in C \setminus \{0\}} wt(v)}$$

ie the weight of the minimum weight non zero code word.

For our example

$C = \langle (0,0,0)^T, (1,1,1)^T \rangle$

and so $d=3$.

We often refer to a linear code using the shorthand

$[n, k, d]$.

In contrast to the dimension, computing the code distance of a linear code is NP-hard.

Tanner graphs

A Tanner (or factor) graph is a convenient representation of the parity-check matrix of a linear code.

Given a pcm H, we add a check node $\square$ to the graph for each row of H and we add a variable node O

to the graph for each column

of H.

Then we connect variable node i

to check j iff $H_{ij} = 1$.

In other words there is an edge

between a check node and

a variable node if the check

acts non-trivially on the bit

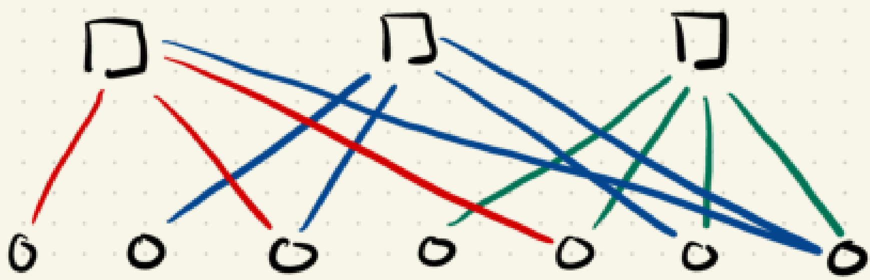
corresponding to the variable

node.

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

(Hamming's code) $[7, 4, 3]$

Tanner graph:



Tanner graphs are often used for **decoding** linear codes.

Given $u \in \mathbb{F}_2^n$ we define the **syndrome** (vector) of u to be $Hu \in \mathbb{F}_2^m$.

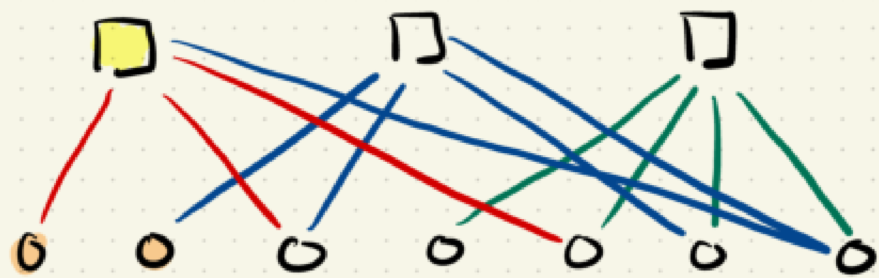
(For a codeword $Hu = 0$ by defn.)

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

$u = (1\,1\,0\,0\,0\,0\,0)^T$

$Hu = (1\,1\,0)$

Graphically



The task of the decoder is to solve the optimization problem

$$\boxed{\underset{u \in \mathbb{F}_2^n \text{ s.t. } Hu = s}{\operatorname{argmin}} \ wt(u)}$$

for a given syndrome s.

The Tanner graph representation is convenient for applying graphical algorithms (e.g. Belief Propagation) to the decoding problem.

Classical LDPC codes

Let $\mathcal{C}$ be a family of linear codes indexed by a parameter L such that the L'th code in the family has parameters $[n(L), k(L), d(L)]$ and pcm H_L.

We say that $\mathcal{C}$ is a <u>good</u> code family if, in the asymptotic limit,

$$k(L) = O(n(L))$$
$$d(L) = O(n(L))$$

We say that $\mathcal{C}$ is an (r,c) LDPC code family if the row weight and column weight of H_L are bounded by r & c, respectively, for all L.

<u>Example</u>: repetition code

We have

$$H_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$H_L = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & & 1 & 1 \end{bmatrix}$$

This is a $(2,2)$ LDPC family.

What are the parameters?

One can show (try it, not hard) that $k(L) = 1$ & $d(L) = n(L)$.

We can therefore express the parameter as $[n(L), 1, n(L)]$.

The repetition code family is LDPC but not good.

There do however exist families of good LDPC codes.

In fact, a randomly generated $m \times n$ matrix w/ constant row and column weight and $m = Rn$ $0 < R < 1$ will w/ high probability be a good code.

These families are used in e.g. WiFi and 5G!

Quantum LDPC codes

The definition is analogous to the classical case.

Let $\{S_L\}$ be a family of stabilizer codes where the L'th code in the family has parameters $[[n(L), k(L), d(L)]]$.

We say that the family is (w, q) LDPC if each stabilizer generator has maximum weight $\leq w$ as each qubit has qubit degree $\leq q$.

We recall that the weight of a Pauli operator is the number of non identity factors in the operator.

e.g. $wt(X1X1) = 2$

$wt(ZZZ1) = 3$

$wt(XYZ) = 3$

For a physical qubit in the code, its qubit degree is the number of stabilizer that act on it. This is analogous to the column weight of a (classical) parity-check matrix.

Example : Quantum repetition code

Stabilizers $\langle ZZ1 \ldots 1, 1ZZ \ldots,$

$1 \ldots 1ZZ \rangle$

$n(L) = L$

$k(L) = 1$

$d(L) = 1$

$[[L, 1, 1]]$ code

(we can think of L as the length of a chain of physical qubits)

$Z1 \ldots 1$ is a logical operator

This is a (2,2) family.

You may have noticed that the LDPC property is not really a property of the code but rather a property of a set of stabilizer generators.

The same is true in the linear code case (stabilizer generators → parity-check matrix)

For a given stabilizer code there are many (exponential) possible sets of stabilizer generators.

So we say that a code is (w,q)-LDPC if there exits a $(w-q)$-LDPC set of stabilizer generators for the code. This is hard to check in the general case!

We focus on the subclass of CSS codes as they are easier to analyse and any non-CSS code can be transformed into a CSS code without changing the scaling of the parameters.

Recall that we can write the stabilizer generators of a CSS code in binary symplectic form

$$H = \left[\begin{array}{c|c} H_X & 0 \\ 0 & H_Z \end{array}\right]$$

where $H_X \in M_{m_X \times n}(\mathbb{F}_2)$

$H_Z \in M_{m_Z \times n}(\mathbb{F}_2)$

and so $H \in M_{m \times 2n}(\mathbb{F}_2)$

where $m = m_X + m_Z$.

This is another way of saying that for a CSS code there exists a set of stabilizer generators consisting of exclusively X-type and Z-type Pauli operators.

Given a set of Pauli operators of a single type, we can represent each operator as an $\mathbb{F}_2$ vector using the mapping $I \to 0$, $P \to 1$.

The commutation condition becomes $\boxed{H_X H_Z^T = 0}$.

For a CSS code we define w_X and q_X to be the max row and column weight of H_X & w_Z and q_Z to be the max row and column weight of H_Z.

Then
$$\boxed{\begin{aligned} w &= \max(w_X, w_Z) \\ q &\leq q_X + q_Z \end{aligned}}$$

Example : Steane's code

$$H_X = H_Z = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(the parity-check matrix of the Hamming code)

$[[7, 1, 3]]$

$w_X = w_Z = 4$ $w = 4$

$q_X = q_Z = 3$ $q = 6$

H_X and H_Z are the parity check matrices of linear codes so we can use a lot of the same tools to study CSS codes.

One can show

$$\boxed{k = n - \operatorname{rank} H_X - \operatorname{rank} H_Z}$$

$$\boxed{d_X = \min_{u \in \ker H_Z / \operatorname{Im} H_X} wt(u)}$$

$$\boxed{d_Z = \min_{u \in \ker H_X / \operatorname{Im} H_Z} wt(u)}$$

row space

$$d = \min(d_X, d_Z)$$

Why do we care about qLDPC codes?

As you will see later in the course, when we consider circuit level error models, the noise associated w/ measuring a stabiliser generally scales w/ its weight.

Therefore, we expect codes w/ low-weight stabilizer generators to have superior performance in practice.

Good qLDPC conjecture

Can a stabilizer code family be both LDPC and good?

Recall: a good code family has $k(L) = O(n(L))$

$d(L) = O(n(L))$

Examples

q Repetition code $k(L) = 1$ $d(L) = 1$ Bad!

Toric code $k(L) = 2$ $d(L) = \sqrt{n(L)}$ Better!

Hypergraph product codes $k(\mathcal{L}) = n(\mathcal{L})$

Even better but still not "good" $d(\mathcal{L}) = \sqrt{n(\mathcal{L})}$

For 20 years the $\sqrt{n}$ distance of the toric code was essentially the best known distance for a qLDPC code.

Building on the hypergraph product construction, there was a flurry of progress in 2020 - 2021 culminating in a paper by Panteleev & Kalachev, who proved that good qLDPC codes exist!

Tanner graphs for CSS codes

We can also draw Tanner graphs for CSS codes. Essentially we combine the Tanner graphs of H_Z & H_X.

Example: $[[8,3,2]]$ code

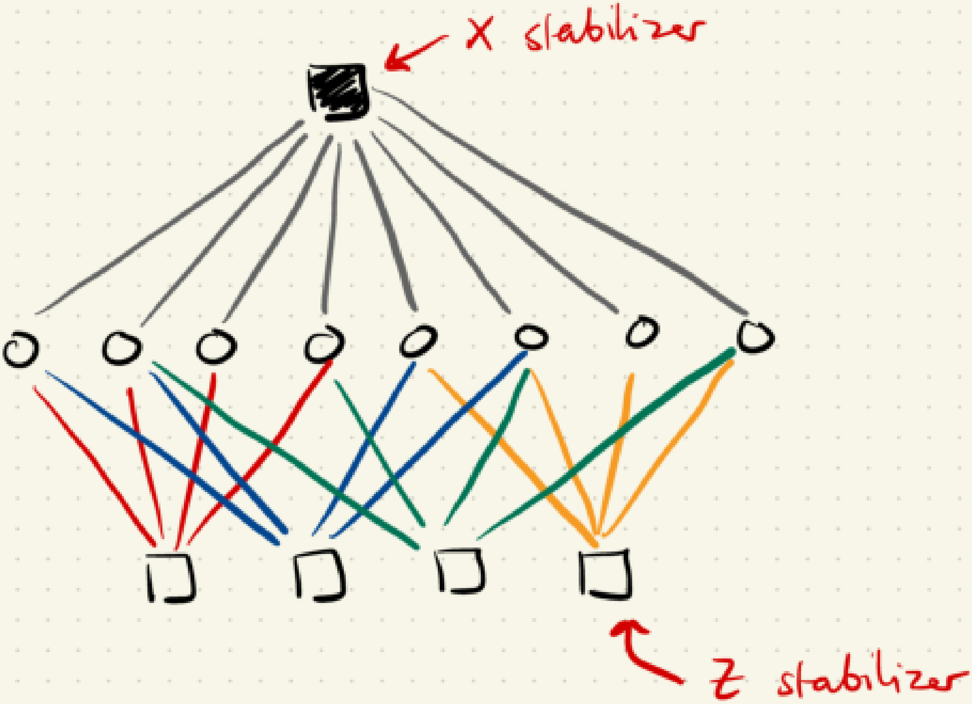
$$H_X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

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

This code was recently used by researchers at Harvard in their breakthrough QEC experiment.

$$H_X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

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



References

- Quantum Low-Density Parity-Check Codes by Breuckmann & Eberhardt, PRX Q 040101, 2021.

- Asymptotically Good Quantum and Locally Testable Classical LDPC codes by Panteleev and Kalachev, STOC 2022.

Warning: not an easy paper!

See video by Ryan O'Donnell explaining the PK construction

https://youtu.be/k7LuOiOBYyQ?feature=shared