Contents

1 Classical linear codes 15.07.22

1.1 Repetition code

Repetition code. Encoding

$$0 \to 000, 1 \to 111 \tag{1}$$

Decoding: majority vote, e.g. $010 \to 0, 011 \to 1$. Succeeds when 0 or 1 bit is flipped, fails when 2 or 3 bits are flipped. Error probability is reduced from p to $3p^2$ (for $p \ll 1$).

1.2 Linear codes

Repetition code is an example of a linear code. A linear code (n, k, d) is defined by a generator matrix G of size $k \times n$ and a parity check matrix H of size $(n-k) \times n$. Code subspace is bG, for all k-bitstrings b. Alternatively, code supspace consists of all n-bitstings b' satisfying $H(b')^T = 0$. For the repetition code

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad H^T G = 0 \tag{2}$$

1.3 Code distance

Code distance is the minimum Hamming distance between any two code words. For a linear code, code distance is the minimum (non-zero) weight of n-bitstrings in the code space. For the repetition code $d = 3$. A code that can correct t errors has distance $d = 2t + 1$.

1.4 Dual code

For a linear code with generator matrix G and parity check matrix H the dual code is another linear code with G and H swapped $G^{\perp} = H, H^{\perp} = G$. For the dual repetition code

$$G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad H = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \quad H^T G = 0 \tag{3}$$

Dual repetition code encodes 2 bits into 3 bits

$$00 \to 000 \tag{4}$$
$$01 \to 011 \tag{5}$$
$$10 \to 110 \tag{6}$$
$$11 \to 101 \tag{7}$$

Distance of the code is 2. It can detect one error, but correct none.

1.5 Hamming bound

Codes can not be ”too good”. Encoding k-bitstrings into n-bitstrings and being able to correct t errors (distance at least $d \geq 2t + 1$) requires the embedding space to have dimension at least

$$2^n \geq 2^k \sum_{i=0}^{t} C_n^i \tag{8}$$

1.6 Gilbert–Varshamov bound

”Good enough” codes exist. Given n physical bits a code with distance d exists encoding k logical bits with

$$2^k \geq \frac{2^n}{\sum_{i=0}^{d-1} C_n^i} \tag{9}$$

2 Quantum repetition code and first look at stabilizer formalism 22.06.2022

2.1 Quantum repetition code

Encoding

$$|0\rangle \to |\bar{0}\rangle = |000\rangle, \qquad |1\rangle \to |\bar{1}\rangle \to |111\rangle, \qquad \alpha|0\rangle + \beta|1\rangle \to \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle \tag{10}$$

Exercise: find a unitary quantum circuit that performs the encoding starting from the state $(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle$.

2.2 X errors

Correctable errors: X_1, X_2, X_3:

$$X_1|000\rangle = |100\rangle, \quad X_1|111\rangle = |011\rangle, \quad \dots \tag{11}$$

2.3 Syndromes

Measuring Z_1Z_2, Z_2Z3 gives syndromes. For correctable errors syndromes are

	Z_1Z_2	Z_2Z_3
Id	1	1
X_1	-1	1
X_2	-1	-1
X_3	1	-1

Excercise: find syndromes of other errors, e.g. X_1X_2 or even Y_1Z_3.

Excercise: find a circuit that uses 1 ancilla qubit to measure syndrome Z_1Z_2.

2.4 Measurement

Measuring Z in state $\alpha|0\rangle + \beta|1\rangle$ gives $+1$ with probability $|\alpha|^2$ and post-measurement state $|0\rangle$ or -1 with probability $|\beta|^2$ and post-measurement state $|1\rangle$.

Exercise: find values, probabilities and post-measurement states of Z_1Z_3 performed on $\alpha|100\rangle + \beta|011\rangle$. What about, say, X_1?

2.5 Necessary and sufficient conditions for error correction

Let $\{|\bar{i}\rangle\}$ be the code space and $\{E_\alpha\}$ the set of errors. The code can correct these errors iff

$$\langle\bar{i}|E_\beta^\dagger E_\alpha|\bar{j}\rangle = 0, \qquad i \neq j \tag{12}$$

$$\langle\bar{i}|E_\beta^\dagger E_\alpha|\bar{i}\rangle = C_{\alpha\beta} \quad \text{independent of } i \tag{13}$$

The first condition means no errors can make different logical states overlap (otherwise they could be confused and the errors could not be corrected). The second means that confusing different errors acting on the same state is fine, as long as the correction procedure works identically on all logical states.

2.6 Shor's code

Shor's code can correct both X and Z single-qubit errors. It uses 5 qubits. Encoding

$$|\bar{0}\rangle = (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \tag{14}$$

$$|\bar{1}\rangle = (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \tag{15}$$

Exercise: propose a syndrome measurement that diagnoses X_1 error and a syndrome that diagnoses Z_1 error without destroying superposition $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$.

3 General properties of stabilizer codes 29.06.2022

3.1 Errors from syndromes

Recall the repetition code and assume that after an error measuring syndromes gives $Z_1 Z_2 = 1$, $Z_2 Z_3 = -1$. What was the error? The error was X_3. Indeed, before error $Z_1 Z_2|\psi\rangle = 1, Z_2 Z_3|\psi\rangle = 1$ and after error $Z_1 Z_2 X_3|\psi\rangle = |\psi\rangle, Z_2 Z_3 X_3|\psi\rangle = -X_3|\psi\rangle$. We can in fact exlude $|\psi\rangle$ from this equation by noting that X_3 is the unique error (among correctable ones) that commutes with $Z_1 Z_2$ and anticommutes with $Z_2 Z_3$. In general, when there are n stabilizers $S_1, \ldots, S_n$ and syndrome is $s_1, \ldots, s_n$ where $s_i = \pm$ our error E (which may not be unique) is the one that satisfies $ES_i = s_i S_i E$.

3.2 Construct code subspace from stabilizers

If $S_1, \ldots, S_n$ is the full set of stabilizers state of the form

$$|\bar{\psi}\rangle = \sum_{i=0}^{n} S_i|\psi\rangle \tag{16}$$

is in code space since $S_j|\bar{\psi}\rangle = |\bar{\psi}\rangle$ for any S_j. Another way to obtain this state is to take a product of projectors

$$|\bar{\psi}\rangle = \prod_g \frac{1+S_g}{2}|\psi\rangle \tag{17}$$

where sum is not over the *generators* of the stabilizer, and not all of the stabilizer group.

3.3 Pauli strings symplectic representation

Any Pauli string can be written as follows

$$P = \pm Z_1^{\alpha_1} \ldots Z_n^{\alpha_n} X_1^{\beta_1} \ldots X_n^{\beta_n} = \pm Z^\alpha X^\beta = (\alpha|\beta) \tag{18}$$

Product

$$(\alpha_1|\beta_1)(\alpha_2|\beta_2) = (-)^{\beta_1\alpha_2}(\alpha_1+\alpha_2|\beta_1+\beta_2) \tag{19}$$

Commutation relations

$$P_1 P_2 = (-)^{\alpha_1\beta_2 - \beta_2\alpha_1} P_2 P_1 \tag{20}$$

3.4 Structure of code

For an $[[n,k,d]]$ stabilizer the group of all Pauli strings of dimension 4^n is divided into (1) Stabilizer space with $n-k$ generators, where all operators commute with each other (2) Space of logical operators with $2k$ generators, which commute with stabilizers but may anticommute with each other and (3) Operators which anticommute with at leas one stabilizer (those are detectable errors) with d generators.

Exercise: $d = ?$.