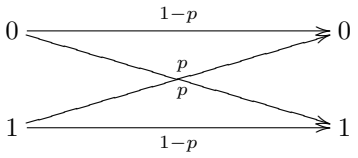
1 Noise Models

Consider the task of transmitting a single bit of classical information through a noisy channel. The noise results in a bit flip with probability $p > 0$, while with probability $1 - p$ the bit is transmitted without any error. Such a channel is known as a binary symmetric channel and is given by



Similarly, we may consider the generalized Pauli channel as an example of a noisy channel for a qubit. The set of errors now consists of Bit Flip X, Phase Flip Z or both $Y = XZ$, with each occurring with probability p_x, p_z and p_y respectively.

$$\rho \to \mathcal{E}(\rho) = p_0\rho + p_x \underbrace{X\rho X}_{\text{Bit Flip}} + p_z \underbrace{Z\rho Z}_{\text{Phase Flip}} + p_y \underbrace{Y\rho Y}_{\substack{\text{Both Errors} \\ Y=ZX}}$$

Now we shall look at protocols that allow us to correct for these errors.

2 Classical 3-bit repetition code

The first thing that comes to mind when considering classical error correcting codes is redundancy, i.e. rather than encoding information in a single bit we encode multiple bits with the same information. As the name implies, the 3-bit repetition code uses three bits to encode a single bit. So we have

$$0 \to 000$$
$$1 \to 111$$

The encoded bit strings 000 and 111 are referred to as logical 0 and logical 1. This encoding protects the input against a single error when it is sent through the binary symmetric channel. The receiver at the end of the channel has to decide whether the original input was 0 or 1 based on the three output bits of the channel. It is not difficult to see that majority voting on the logical bits recovers the original bit. This happens with probability $1 - 3p^2 + p^3$. Recall that the original error probability was p which now goes down whenever $p < 1/2$.

3 Quantum Error Correction

It is natural to consider whether a similar repetition code could work for quantum information. However, the following issues need to be dealt with before we can make progress:

Input	Channel Output	1^{st} bit $= 2^{nd}$ bit	1^{st} bit $= 3^{rd}$ bit
000	000	Y	Y
	001	Y	N
	010	N	Y
	100	N	N
111	111	Y	Y
	110	Y	N
	101	N	Y
	011	N	N

Table 1: Single Error possibilities for the 3-bit repetition code

1. *No Cloning*: We cannot copy quantum information, i.e. we cannot have:
$$|\psi\rangle = a|0\rangle + b|1\rangle \rightarrow |\psi\rangle^{\otimes 3}$$
2. *Handling Measurements*. If we try $|\psi\rangle = a|000\rangle + b|111\rangle$ as a possible encoding, then measuring the output state in order to determine the error would generally also kill the state.

3.1 3-qubit bit flip code

We want to build a code that protects against a bit flip, i.e. with probability p the transmitted qubit $|\psi\rangle$ is taken to $X|\psi\rangle$. Let $|\psi\rangle = a|0\rangle + b|1\rangle$. We perform the following encoding:

$$\begin{aligned} |0\rangle &\rightarrow |0\rangle_L = |000\rangle \\ |1\rangle &\rightarrow |1\rangle_L = |111\rangle \\ |\psi\rangle &\rightarrow a|0\rangle_L + b|1\rangle_L = a|000\rangle + b|111\rangle \end{aligned}$$

Now, there are four possible situations that we need to take care of:

$$\begin{aligned} |\psi\rangle_L &\xrightarrow{\text{No Error}} a|000\rangle + b|111\rangle \\ &\xrightarrow{X_3} a|001\rangle + b|110\rangle \\ &\xrightarrow{X_2} a|010\rangle + b|101\rangle \\ &\xrightarrow{X_1} a|100\rangle + b|011\rangle \end{aligned}$$

Measuring the eigenvalues of the observables $\{ZZI, IZZ\}$ we can perform a parity check to determine with certainty which qubit has been flipped hence allowing us to recover the original state. Note that this measurement does not destroy the logical state.

3.2 3-qubit phase flip code

We want to build a code that protects against a phase flip, i.e. with probability p the transmitted qubit $|\psi\rangle$ is taken to $Z|\psi\rangle$. Note that that phase flip error is similar to the bit flip error if we work in the $\{|+\rangle, |-\rangle\}$ basis ($Z|+\rangle = |-\rangle$ and vice versa), where

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

We perform the following encoding:

$$|\psi\rangle_L = a|+++\rangle + b|---\rangle$$

Measuring the observables $\{XXI, IXX\}$ allows us to recover the original state.

4 9-bit Shor code

The Shor code protects a single qubit against any single arbitrary error. The idea is simple – we basically *concatenate* the previous two codes resulting in a more robust error correcting scheme. By concatenation we basically mean that we perform one encoding on top of another. First we encode for the phase flip error and then encode the resulting state for the bit flip error. The procedure proceeds as follows

$$\begin{aligned} \text{Original State } |\psi\rangle &= a\,|0\rangle + b\,|1\rangle \\ \text{Inner Code } |\psi\rangle_{L_1} &= a\,|+++\rangle + b\,|---\rangle \\ \text{Outer Code } |\psi\rangle_{L_2} &= a(|000\rangle + |111\rangle)^{\otimes 3} + b(|000\rangle - |111\rangle)^{\otimes 3} \end{aligned}$$

The decoding procedure allows us to recover the original state if only a single bit flip or phase flip error has occurred. When we decode we first check and correct for a bit flip and then perform the correction on the resulting state for phase flip. It should be noted that order matters in this process. For the above choice of logical encoding, the inner code needs to be for the phase flip error and the outer code for bit flip. The protocol does not work if we reverse the order.

5 Classical Binary Linear $[n, k, d]$ Code

Definition 1 *A binary linear $[n, k, d]$ code C is a linear subspace of dimension k (2^k codeword possibilities) of $\mathbb{Z}_n^2$, such that the minimum Hamming weight of non-zero codewords is d.*

The key idea behind our selection of codewords is the notion that noisy codewords do not overlap, hence the choice for distance d results in a $[n, k, d]$ code C being able to correct for up to $\frac{d-1}{2}$ errors. The code space C is given by the generator matrix $G \in \mathbb{Z}_2^{k\times n}$. We may define the parity check matrix $H \in \mathbb{Z}_2^{[n-k]\times n}$ such that $HG^T = 0$, i.e. for any word in the code space if we do a parity check we obtain 0.

Example 2 $[3, 1, 3]$ *code*

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$
$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Example 3 $[7, 4, 3]$ *Hamming code*

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$
$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The lecture concluded with a brief preview of CSS codes from the next lecture.