Lecture 17: General quantum errors; CSS codes

March 23, 2006

General quantum errors

In the previous lecture we discussed Shor's 9 qubit code, which can be viewed as an encoding of one qubit into three blocks of three qubits. It protects against up to 1 bit-flip error in each block and any number of phase-flip errors contained in any one block. In particular, it can protect against any one of the three errors

$$\sigma_x,\ \sigma_z,\ \sigma_x\sigma_z$$

occurring on a single qubit. The code of course does not require that an error takes place to work correctly, so we can add the identity matrix to the above list. Finally, because $\sigma_y = -i\sigma_x\sigma_z$ and global phase factors are irrelevant, we may say that Shor's 9 qubit code protects against any of the four possible errors

$$I,\ \sigma_x,\ \sigma_y,\ \sigma_z$$

on any one qubit.

It turns out that this is enough to conclude that the code protects against an *arbitrary* one-qubit error, represented by any single-qubit admissible operation, and the same may be said of any quantum error correcting code. The first part of this lecture will be devoted to demonstrating this fact. The situation generalizes to multiple-qubit errors and codes that can protect against multiple qubit errors, but for simplicity we will focus on the single-qubit case.

Suppose A is any 2×2 complex matrix. Then

$$A = aI + b\sigma_x + c\sigma_y + d\sigma_z$$

for some choice of $a, b, c, d \in \mathbb{C}$. Forget about the fact that A may not be unitary for a moment, and imagine that A is applied to the j-th qubit in some n qubit state $|\psi\rangle$, which we will view as encoding one or more qubits in some quantum error correcting code. We obtain

$$A^{(j)}|\psi\rangle = a\ |\psi\rangle + b\,\sigma_x^{(j)}|\psi\rangle + c\,\sigma_y^{(j)}|\psi\rangle\, d\,\sigma_z^{(j)}|\psi\rangle\,,$$

where the superscript (j) indicates which qubit each mapping acts on.

Assuming that the code protects against any one of the four errors given by the Pauli matrices, the first step of error correction would result in the state

$$\begin{aligned} & a\ |\psi\rangle\,|I \text{ syndrome}\rangle \\ +& b\,\sigma_x^{(j)}|\psi\rangle\,|\sigma_x^{(j)} \text{ syndrome}\rangle \\ +& c\,\sigma_y^{(j)}|\psi\rangle\,|\sigma_y^{(j)} \text{ syndrome}\rangle \\ +& d\,\sigma_z^{(j)}|\psi\rangle\,|\sigma_z^{(j)} \text{ syndrome}\rangle\,. \end{aligned}$$

Correcting according to the syndrome then gives

$$|\psi\rangle \left(a \ |I \text{ syndrome}\rangle + b \ |\sigma_x^{(j)} \text{ syndrome}\rangle + c \ |\sigma_y^{(j)} \text{ syndrome}\rangle + d \ |\sigma_z^{(j)} \text{ syndrome}\rangle\right)$$

Alternately, if the syndrome is measured and the error is corrected appropriately, the state $|\psi\rangle$ is recovered regardless of the measurement outcome.

Now, if an arbitrary one-qubit *admissible* operation occurs on qubit j, we obtain

$$\Phi^{(j)}(|\psi\rangle\langle\psi|) = \sum_{k=1}^{N} A_k^{(j)} |\psi\rangle\langle\psi| \left(A_k^{(j)}\right)^\dagger$$

for some collection of matrices $A_1^{(j)}, \ldots, A_N^{(j)}$, which can each be written

$$A_k^{(j)} = a_k I + b_k \sigma_x^{(j)} + c_k \sigma_y^{(j)} + d_k \sigma_z^{(j)}.$$

A slightly messy (but conceptually simple) calculation shows that if the syndrome is computed and measured, we obtain the mixed state

$$\begin{aligned}\sum_{k=1}^{N} \Big(& |a_k|^2 |\psi\rangle\langle\psi| \otimes |I \text{ syndrome}\rangle\langle I \text{ syndrome}| \\ & |b_k|^2 \sigma_x^{(j)} |\psi\rangle\langle\psi| \sigma_x^{(j)} \otimes |\sigma_x^{(j)} \text{ syndrome}\rangle\langle\sigma_x^{(j)} \text{ syndrome}| \\ & |c_k|^2 \sigma_y^{(j)} |\psi\rangle\langle\psi| \sigma_y^{(j)} \otimes |\sigma_y^{(j)} \text{ syndrome}\rangle\langle\sigma_y^{(j)} \text{ syndrome}| \\ & |d_k|^2 \sigma_z^{(j)} |\psi\rangle\langle\psi| \sigma_z^{(j)} \otimes |\sigma_z^{(j)} \text{ syndrome}\rangle\langle\sigma_z^{(j)} \text{ syndrome}| \Big).\end{aligned}$$

The fact that there are no cross-terms in this expression is a consequence of measuring the syndrome (with respect to the standard basis).

Although the above expression might look unappealing to you, it is expressing something quite remarkable. The act of computing the syndrome and measuring has effectively projected the arbitrary error represented by Φ into one of the four discrete errors I, σ_x, σ_y, or σ_z. Correcting according to the syndrome then gives

$$\begin{aligned}|\psi\rangle\langle\psi| \otimes \sum_{k=1}^{N} \Big(& |a_k|^2 |I \text{ syndrome}\rangle\langle I \text{ syndrome}| \\ & + |b_k|^2 |\sigma_x^{(j)} \text{ syndrome}\rangle\langle\sigma_x^{(j)} \text{ syndrome}| \\ & + |c_k|^2 |\sigma_y^{(j)} \text{ syndrome}\rangle\langle\sigma_y^{(j)} \text{ syndrome}| \\ & + |d_k|^2 |\sigma_z^{(j)} \text{ syndrome}\rangle\langle\sigma_z^{(j)} \text{ syndrome}| \Big).\end{aligned}$$

Here we are of course relying on the fact that the code corrects the four Pauli operators correctly. Throwing the syndrome in the trash gives $|\psi\rangle\langle\psi|$ as desired.

CSS codes

The last thing we will discuss on the topic of quantum error correction is CSS codes, named after their co-discoverers Robert Calderbank, Peter Shor, and Andrew Steane. These codes are interesting because they illustrate how classes of classical error correcting codes can sometimes be turned into classes of quantum codes. They are also useful for one of the simpler proofs of security of the BB84 quantum key-distribution protocol, which we will discuss next week.

Classical linear codes

In order to discuss CSS codes, we will need to first learn a little bit about classical linear codes. Some terminology will be helpful when doing this.

First, when we refer to an $[n, k]$ code, we mean an error correcting code that encodes k bits into n (so we will always have $n > k$). Formally, we define an $[n, k]$ code simply to be a subset of $\mathbb{Z}_2^n$ of size 2^k. Here, and throughout this discussion, we are identifying strings of length n with (column) vectors in $\mathbb{Z}_2^n$ in the most straightforward way. The elements of such a code $C \subseteq \mathbb{Z}_2^n$ are called *codewords*.

An $[n, k]$ code C is said to have distance d if the *Hamming distance* between any two distinct code words $y, z \in C$ is at least d, and d is the largest positive integer that satisfies this property. The Hamming distance between strings y and z is the number of positions in which they disagree. An $[n, k]$ code with minimum distance d is sometimes referred to as an $[n, k, d]$ code. Such a code can correct bit-flips on any number of bits that is strictly less than half of the code's minimum distance. An $[n, k]$ *linear code* is a code that happens to form a subspace of $\mathbb{Z}_2^n$ of dimension k, where all arithmetic takes place in $\mathbb{Z}_2$. In the case of a linear code, the minimum distance is the same as the minimal *Hamming weight* (or number of 1s) of any nonzero codeword.

There are two matrices that can be associated with a linear code that make encoding and error detection/correction easier: a *generator matrix* and a *parity check matrix*.

Generator matrix. A generator matrix G for an $[n, k]$ linear code C is an $n \times k$ matrix over $\mathbb{Z}_2^n$ such that

$$C = \operatorname{range}(G) \stackrel{\text{def}}{=} \{Gx \,:\, x \in \mathbb{Z}_2^k\}.$$

To encode a string $x \in \mathbb{Z}_2^k$, we simply multiply by G, i.e., $x \in \mathbb{Z}_2^k$ is encoded as $Gx \in \mathbb{Z}_2^n$.

Parity check matrix. A parity check matrix K for an $[n, k]$ linear code C is an $(n-k) \times n$ matrix over $\mathbb{Z}_2$ such that

$$C = \ker(K) \stackrel{\text{def}}{=} \{y \in \mathbb{Z}_2^n \,:\, Ky = 0\}.$$

The syndrome for a given string $y' \in \mathbb{Z}_2^n$ is $Ky' \in \mathbb{Z}_2^{n-k}$.

Once either of these two matrices is chosen, the code is determined. It is helpful, however, to have both.

Let us briefly discuss why the syndrome given by the parity check matrix is helpful. Suppose that we have a codeword $y \in C$ and errors occur on one or more of the bits. We can write the resulting string y' as $y + e$ where $e \in \mathbb{Z}_2^n$ indicates where the errors took place (a 1 in each position where an error takes place, and 0 everywhere else). Then we see that $Ky' = K(y + e) = Ke$ is

a function only of the errors and not the original codeword. If the Hamming weight of e is small enough, the syndrome Ke will uniquely determine the error vector e. More specifically, there cannot be two distinct strings e and e' both of Hamming weight less than $d/2$ such that $Ke' = Ke$.

Example 1. Let us define a $[7, 4]$ linear code C as follows. The generator matrix is

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

and the parity check matrix is

$$K = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

This particular code is called a $[7, 4]$ Hamming code. (More generally, Hamming codes are $[2^m - 1, 2^m - m - 1]$ codes obtained by taking the parity check matrix to have as columns all of the nonzero strings of length $n - k$.) All Hamming codes including the above one have distance 3. Because the distance is 3, the code can tolerate up to one bit-flip.

For example, the encoding of the string 1001 is 100100 and the encoding of 1111 is 1111111, because

$$G \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{and} \qquad G \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Dual codes

Suppose that we have an $[n, k]$ linear code C. The *dual code* to C is denoted $C^\perp$, and contains all strings that have modulo 2 inner product 0 with all codewords of C. In symbols:

$$C^\perp = \{y \in \mathbb{Z}_2^n \,:\, y \cdot x = 0 \text{ for all } x \in C\}.$$

Therefore, $C^\perp$ is an $[n, n - k]$ linear code. If we have a generator matrix G and a parity check matrix K for C, then the generator matrix for $C^\perp$ is K^T and the parity check matrix is G^T. A code C is *weakly self dual* if $C \subseteq C^\perp$ and is *strictly self dual* (or just *self dual*) if $C = C^\perp$.

Example 2. The code $C = \{00, 11\}$ is a very simple self dual $[2, 1]$ linear code. The $[7, 4]$ Hamming code above is not weakly self dual. In order for C to be weakly self dual, it is necessary and sufficient that the generator matrix G for C satisfies $G^{\mathsf{T}}G = 0$. This is not the case for the $[7, 4]$ Hamming code. However, the dual code $C^{\perp}$ of that code is weakly self dual. This follows from the fact that the generator matrix of $C^{\perp}$ is K^{T}, which satisfies $(K^{\mathsf{T}})^{\mathsf{T}}K^{\mathsf{T}} = KK^{\mathsf{T}} = 0$.

CSS codes

Now we are ready to define CSS codes. In order to define a CSS code, we must have two classical linear codes C_1 and C_2 that satisfy certain properties:

1. C_1 must be an $[n, k_1]$ linear code and C_2 must be an $[n, k_2]$ linear code for $k_2 < k_1$.
2. It must be that $C_2 \subseteq C_1$.
3. If both C_1 and $C_2^{\perp}$ can correct up to t errors, then the resulting CSS code will be an $[n, k_1 - k_2]$ code that corrects up to t quantum errors (meaning an arbitrary error confined to t qubits).

Example 3. Let C_1 be the $[7, 4]$ Hamming code discussed above and let $C_2 = C_1^{\perp}$. Then C_2 is a $[7, 3]$ (weakly self dual) linear code. We have $C_2 \subsetneq C_1$ by virtue of the fact that C_2 is weakly self dual and is clearly not equal to C_1 (because it has fewer vectors). Because C_1 corrects up to one error and $C_2^{\perp} = C_1$, the CSS code constructed will be a $[7, 1]$ quantum code that can correct 1 quantum error.

The encoding for the CSS code constructed from codes C_1 and C_2 as above is as follows.

1. Let $N = 2^{k_1 - k_2}$. The space spanned by all possible encodings will have dimension N. Choose codewords $x_0, \ldots, x_{2^N - 1} \in C_1$ satisfying the condition

   $$x_i + x_j \notin C_2$$

   for $i \neq j$. (This will always be possible because C_1/C_2 is a space of dimension $k_1 - k_2$ over $\mathbb{Z}_2$, and each element of this space gives rise to at least one good choice of x_j.)
2. Identify the classical states of the $k_1 - k_2$ qubits to be encoded with numbers $0, \ldots, N - 1$ in binary. Then the encoding corresponds to the linear mapping defined by

   $$|j\rangle \mapsto |x_j + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x_j + y\rangle .$$

   The fact that $\langle x_i + C_2 | x_j + C_2 \rangle = 0$ for $i \neq j$ follows from the fact that $x_i + x_j \notin C_2$ for $i \neq j$.

Example 4. Let C_1 and C_2 be as in the previous example. Then $N = 2$, so we need just two codewords x_0 and x_1 in C_1 so that $x_0 + x_1 \notin C_2$. We may take $x_0 = 0000000$ and $x_1 = 1111111$. Enumerating all elements of $C_2 = C_1^{\perp}$ is not so hard in this case—we have

$$C_2 = \{0000000, 0001111, 0110011, 1010101, 0111100, 1011010, 1100110, 1101001\}.$$

Thus, the encoding for the corresponding CSS code will be:

$$\begin{aligned}|0\rangle \mapsto \frac{1}{\sqrt{8}}(&|0000000\rangle + |0001111\rangle + |0110011\rangle + |1010101\rangle \\ &+ |0111100\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle) \\ |1\rangle \mapsto \frac{1}{\sqrt{8}}(&|1111111\rangle + |1110000\rangle + |1001100\rangle + |0101010\rangle \\ &+ |1000011\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle).\end{aligned}$$

This code is known as the Steane 7 qubit code.

It remains to discuss error correction for CSS codes. Suppose we have an n qubit state contained in a register R that we want to correct. The procedure for doing this is very simple:

1. First, reversibly compute the syndrome of R for code C_1, which corresponds to the reversible transformation
$$|y\rangle |00\cdots 0\rangle \mapsto |y\rangle |s_1(y)\rangle ,$$
where $s_1(y)$ denotes the syndrome of y with respect to code C_1. Measure this syndrome, and correct bit-flips by applying NOT gates to the appropriate qubits of R.
2. Apply a Hadamard transform to every qubit of R.
3. Repeat the same procedure as in step 1, except using the code $C_2^{\perp}$.
4. Again apply Hadamard transforms to all qubits of R.

Under the assumption that at most t bit-flips and t phase-flips occurred on X starting from some valid encoding $|\psi\rangle$, the result will be that the encoding $|\psi\rangle$ is recovered. Let us see why this is the case.

First, some notation is needed. Suppose that bit-flips are represented by a vector $e \in \mathbb{Z}_2^n$ and phase-flips are represented by $f \in \mathbb{Z}_2^n$, with both vectors containing at most t ones. If, for a given vector $v \in \mathbb{Z}_2^n$ we denote $X^v = X^{v[1]} \otimes \cdots \otimes X^{v[n]}$ and similarly $Z^v = Z^{v[1]} \otimes \cdots \otimes Z^{v[n]}$, this means that the overall error is given by $X^e Z^f$. It will be helpful to note that for any choice of $u, v \in \mathbb{Z}_2^n$ the following simple formulas hold:

(a) $X^u Z^v = (-1)^{u \cdot v} Z^v X^u$,

(b) $H^{\otimes n} X^u = Z^u H^{\otimes n}$, and

(c) $H^{\otimes n} Z^u = X^u H^{\otimes n}$.

Now, consider an arbitrary valid encoded state with respect to the CSS code:

$$\sum_{j=0}^{N-1} \alpha_j |x_j + C_2\rangle .$$

If the errors represented by e and f occur, the state becomes

$$\sum_{j=0}^{N-1} \alpha_j X^e Z^f \left| x_j + C_2 \right\rangle = \sum_{j=0}^{N-1} \alpha_j (-1)^{e \cdot f} Z^f \left| x_j + e + C_2 \right\rangle .$$

Step (1) of the error correction procedure computes and measures the syndrome with respect to C_1. Error correction is performed by applying NOT gates to the qubits indicated by the syndrome. Because e has at most t ones, the syndrome reveals the vector e, and after correction the state becomes

$$\sum_{j=0}^{N-1} \alpha_j (-1)^{e \cdot f} X^e Z^f \left| x_j + C_2 + e \right\rangle = \sum_{j=0}^{N-1} \alpha_j Z^f \left| x_j + C_2 \right\rangle .$$

Notice that here is where we require that $C_2 \subseteq C_1$—it must be that every string in $x_j + C_2$ is a codeword with respect to C_1. It will be convenient for the next step to write the above state as

$$\sum_{j=0}^{N-1} \alpha_j Z^f X^{x_j} \left| C_2 \right\rangle .$$

Next, the Hadamard transforms are performed. It is a straightforward calculation to show that $H^{\otimes n} \left| C_2 \right\rangle = \left| C_2^{\perp} \right\rangle$, and so the state becomes

$$\sum_{j=0}^{N-1} \alpha_j H^{\otimes n} Z^f X^{x_j} \left| C_2 \right\rangle = \sum_{j=0}^{N-1} \alpha_j X^f Z^{x_j} \left| C_2^{\perp} \right\rangle = \sum_{j=0}^{N-1} \alpha_j (-1)^{f \cdot x_j} Z^{x_j} \left| f + C_2^{\perp} \right\rangle .$$

Step (3) of the error correction procedure detects the vector f in a similar way to e being detected in step (1). The correction works by applying NOT gates in the positions indicated by f, which results in the state

$$\sum_{j=0}^{N-1} \alpha_j (-1)^{f \cdot x_j} X^f Z^{x_j} \left| f + C_2^{\perp} \right\rangle = \sum_{j=0}^{N-1} \alpha_j Z^{x_j} \left| C_2^{\perp} \right\rangle .$$

Finally, the second collection of Hadamard transforms map this state to

$$\sum_{j=0}^{N-1} \alpha_j H^{\otimes n} Z^{x_j} \left| C_2^{\perp} \right\rangle = \sum_{j=0}^{N-1} \alpha_j X^{x_j} \left| C_2 \right\rangle = \sum_{j=0}^{N-1} \alpha_j \left| x_j + C_2 \right\rangle .$$

This is the original encoding as required.