Lesson I

Error Correcting Codes

Some Concepts

Z_2 denotes the integers mod 2
$Z_2 = \{0, 1\}$ and addition and multiplication is mod 2
$1 + 1 = 0 \bmod 2 \qquad 1 \cdot 1 = 1 \bmod 2$

Z_2^n stands for the vector space of n-tuples mod 2

$Z_2^3 = \{(0,0,0), (1,0,0), (0,1,0), (0,0,1)$
$(1,1,0), (1,0,1), (0,1,1), (1,1,1)$
Addition is mod 2
$(1,1,0) + (0,1,1) = (1,0,1)$

The number of vectors in Z_2^n is 2^n. The dimension is n.

Error Correcting Codes

A message is an n-tuple in Z_2^n. Suppose it is c. c is sent and some digits may be accidently changed. Suppose r is received. Then $r = c + e$ where e is the error vector. The object is to find c from r.

Usually we find e and then compute $c = r + e$ (remember we are working mod 2 so $e = -e$.

Any subset C of Z_2^n can be used as the ~~space~~ set of codes. Some C are much better to use than others. If C is a subspace, the the code is called a linear code. In that case $|C| = 2^{\dim(C)}$

Two fundamental questions, given r and C

1. Is $r \in C$, a code word
2. If $r \notin C$, what $c \in C$ is closest to r, if one c exists.

Then we assumed that c is the code that was sent.

1. A simple way to check one, if feasible, is to check if $r = c$ for some $c \in C$
2. For 2 one assumes the nearest neighbor policy.

Given $x = (x_1, \ldots x_n)$ and $y = (y_1, \ldots y_n)$

Let $d(x,y) = \sum |x_i - y_i|$. $d(x,y)$ is called the Hamming distance from x to y. If there is a smallest $d(x,c)$, $c \in C$, then c is the closest vector to x and we assume that c is the vector that was sent. The problem with this simple method is the number of c's that need to be checked.

Introduce, for integer t, $B_t(x) = \{y \in V;\ d(x,y) \leq t\}$, a ball of radius t centered at x. Also let d be the shortest distance between vectors in C. If $t < \frac{d}{2}$, then $B_t(c_1) \cap B_t(c_2) = \phi$ for all $c_1, c_2 \in C$. Thus if t errors have been made to get r, r can be in only one ball of radius t around code words. Thus r corrects to that c in the center of the ball.

The r can be corrected if no more than t errors have been made. d is called the designed distance of the code. Designed because we construct the code so that it's distance is d.

Example $C = \{(1100000), (0111110), (0001011)\}$. $d = 3$, hence $t = 1$. If $r = (0110100)$, the distance between r and (0111110) is 1, so r corrects to (0111110)

Example $C = \{(00000000), (11100011), (00011111), (11111100)\}$ what are d and t? Correct $r = (11010011)$

In constructing a code, we would like both C and t to be as large as possible. These ideas conflict. For consider $B_t(c)$. The number of vectors that differ from c in j positions is $\binom{n}{j}$ where we are working in $\mathbb{Z}_2^n$. Thus the number of vectors in $B_t(c)$ is $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$. Since $B_t(c) \cap B_t(c') = \phi$ for all $c, c' \in C$, there are $|C| \left[\binom{n}{0} + \cdots + \binom{n}{t}\right]$ vectors in the $|C|$ balls. Since $|\mathbb{Z}_2^n| = 2^n$,

$$|C| \left[\binom{n}{0} + \cdots + \binom{n}{t}\right] \leq 2^n$$

This inequality is called the Hamming bound.

As you can see, if $|C|$ increases, then t likely decreases and conversely.

Ex. In the last example, $n=8$, $|C|=4$. Compute $\binom{8}{0}+\binom{8}{1}+\binom{8}{2} = 37$. Then $|C| \cdot 37 = 148 < 2^8 = 256$. So we could hope to correct $t=2$ errors. Adding $\binom{8}{3}$ shows that 3 errors can not be corrected.

A word on what we just did. The Hamming bound says a code which satisfies the inequality might be able to correct t errors. However, it is not guaranteed that we will be able to correct t errors. The bound is a necessary but not sufficient condition.

Also note that there are $256-148$ vectors that are not in any of the balls, so are not useful for correction. So our wish list is now that

1. $|C|$ is large
2. t is large
3. The number of vectors not in any ball is small.

If every vector is in one of the balls, then the code is called a perfect code. They are rare. We will soon see an example of perfect codes, the Hamming codes.

~~Finding b~~

Finding balls that come close to covering a space is a famous mathematical problem, called the sphere packing problem.