ERROR CORRECTIN CODES

CHAPTER 1 12 pages

1. Introduction
7. REED MULLER CODES

CHAPTER 2 15 pages

1. Linear Codes
13 Hamming Codes

CHAPTER 3 BCH CODES 15 pages

1. FINITE FIELDS
6 BCH CODES CONSTRUCTION
9 BCH CODES CORRECTION

ERROR CORRECTING CODES

CHAPTER 1

General Codes

Introduction

Z_2 will denote the ring of integers mod 2 This $Z_2 = \{0,1\}$ and $-1 = 1$. Z_2^n will denote the n dimensional vectors space of n-tuples over Z_2.

Error correcting codes are introduced This is when an electronic message c is sent and possibly some digits are accidently changed so that r is received. The challenge is to find the errors and correct them back to c.

The number of vectors in Z_2^n is 2^n. and the dimension is n. Any subset C of $V = Z_2^n$ can serve as the set of codes If C is a subspace of V, then C is called a linear code In that case $|C| = 2^{\dim C}$.

Two fundamental questions are considered

1. Given a vector x, is $x \in C$
2. If x is not in C, what is the vector $c \in C$ that is closest to x, if there is just one.

In principal, the first of these questions could be answered by running through C to see if x appears. The second uses the nearest neighbor policy. Thus let $x = (x_1, \cdots, x_n)$ and $c = (c_1, \cdots c_n)$

Define $d(x,c) = \sum_{i=1}^{n} |x_i - c_i|$. $d(x,c)$ is called the Hamming distance from x to c. If there is a smallest $d(x,c)$, then c is the closest vector to x.

To aid us, for a given integer t, let $B_t(x) = \{y \in V / d(x,y) \leq t\}$

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 each c_1 and c_2 in C. Thus if ~~d t~~, t errors have been made when c is sent, the result $r \in B_t(c)$ and in no other $B_t(c_1)$. So r corrects to c. Hence t errors can always be corrected when $t < \frac{1}{2} d$. d is called the designed distance of C.

Example. $C = \{(110000), (011110), (010101)\}$

Comparing distance between those three vectors, $d = 4$. Hence $t = 1$.

If $r = (011010)$, the distance between r and (011110) is 1 so r is corrected to (011110)

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. Those ideas conflict. For consider $B_t(c)$. The number of vectors that differ from c in j positions in $\binom{n}{j}$ (n is the length of the vectors). Thus the number of vectors in $B_t(c)$ is $\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{t}$. If $B_t(c)\cap B_t(c')=\phi$ for all $c, c' \in C$, then there are $|C|\left(\binom{n}{0}+\cdots+\binom{n}{t}\right)$ vectors in the $|C|$ balls. Since $|V|=2^{\dim V}=2^n$,

$$|C|\left(\binom{n}{0}+\cdots+\binom{n}{t}\right)\le 2^n.$$

This is called the Hamming bound.

Ex In the last example, $n=8$, $|C|=4$. Compute $\binom{n}{0}+\binom{n}{1}+\binom{n}{2}=\binom{8}{0}+\binom{8}{1}+\binom{8}{2}=1+8+28=37$. Then $|C|[1+8+28]=148<2^8=256$. So it might be possible to correct $t=2$ errors. Adding $\binom{8}{3}$ into

the mix 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. It is not guaranteed. What is guaranteed in the inequality is that if it fails, we can not correct t errors. Also note that there are 256 - 148 vectors that are not in any of the balls so there is no guarantee that ~~the~~ any one of them can be corrected; ie, is closest to exactly one member of C. So the vectors space has unused vectors. We could add to our wish list

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; ie, no wasted vectors, then the code

is called perfect. We will soon see an example of a perfect code, the Hamming codes. Finding balls that come close to covering a space is a famous mathematical problem, the sphere packing problem.