Recitation 7

9-29-2011

Today: ① Linear Block Codes (simplifies encoding/decoding tremendously!)
② Rectangular (Product) codes (Parity-check)

Last Time:
Channel Coding (ECC) — Two key ideas!

① Embeddings:
~ place msgs geometrically in larger dimensional space to allow better distance between valid codewords

② Parity Calculations:
~ Compute linear (algebraic) functions of D_i's (data bits) in msg!

Defns:

① Hamming Distance ~ btn two codewords $\vec{v}$ and $\vec{w}$ is the number of coordinates in which they differ.

i.e. $HD(\vec{v},\vec{w}) = d_H(\vec{v},\vec{w}) = \left|\{i \mid v_i \neq w_i,\ i=0,1,\dots n-1\}\right|$

(Useful in determining code's error detection & correction capability)

eg $\left.\begin{matrix}\vec{v}=1\,0\,0\,1 \\ \vec{w}=1\,1\,0\,0\end{matrix}\right\}$ $HD(\vec{v},\vec{w}) = 2$

② Weight (w) ~ of a codeword is the number of non-zero coordinates in the codeword.

eg $\vec{c} = (1\,0\,0\,1\,1\,0\,1\,1)$

$w(\vec{c})$ = weight = 5

(3) <u>Minimum Distance</u> (d_{min}) of a block code

~ min Hamming distance btn all distinct pairs of codewords in $\mathcal{C}$.

$$d_{min} = \min\{d_H(\vec{v}_i, \vec{v}_j)\} \quad \forall\, i \neq j.$$

For an error to be ~~de~~ <u>undetectable</u>, it must change the symbol values in at least d_{min} coordinates for one codeword to look like another in $\mathcal{C}$;

ie. <u>Thm</u>: A code with d_{min} can detect all error patterns of weight less than or equal to ($d_{min} - 1$)

$$\underline{t_{detect} \leq d_{min} - 1}.$$

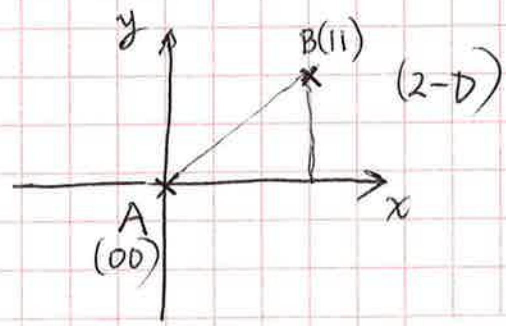
(*) Consider, for example:
Want to send {Apple (A), Banana (B)}

<u>Case I</u>:

A — 0
B — 1 } Distance btn codewords = 1
Thus <u>cannot detect</u> anything!

<u>Case II</u>:

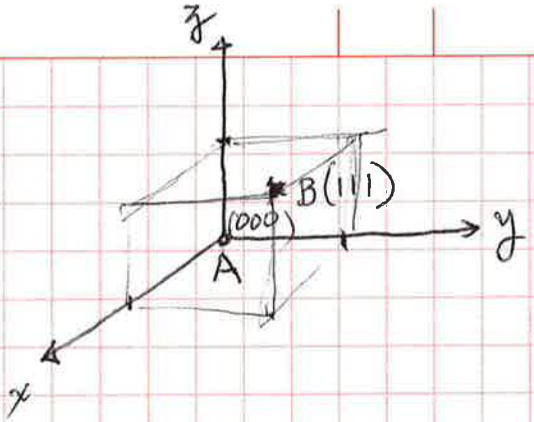
A — 00
B — 11 } Distance btn codewords = 2
Thus, can <u>detect single error</u>
(eg. 10 or 01 can detect, but <u>cannot correct</u>!)



Case III: A — 000, B — 111 } Distance b/n codewords = 3

Thus can detect up to 2 errors

Can correct 1 error!



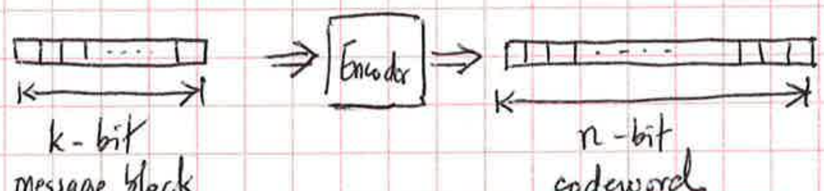
Continuing our observation, it can be seen that

$$\boxed{t_{detect} \le d_{min} - 1}$$ as before.

Also, Thm: $$\boxed{t_{correct} \le \left\lfloor \frac{d_{min}-1}{2} \right\rfloor}$$

Linear Block Codes

Recall: a (n,k) block code of length n contains 2^k codewords, where k is the length of the original message.



Since there are 2^n possible n-bit words, there are $(2^n - 2^k)$ invalid ~~codes~~ words, ie $(2^n - 2^k)$ words not associated with codewords.

aka, the codebook contains redundancy!

Redundancy $r = n - k$

Code Rate $R = \frac{k}{n}$

Making code linear simplifies certain properties & makes implementation easier!

Properties of Linear Blk Codes:

① $\vec{c}_i + \vec{c}_j \in \mathcal{C}$ (any linear combination of codewords is a codeword)

~ as a result, a linear code always contains the all-zero vector!

(eg1) Is $\mathcal{C} = \{(0000), (1111)\}$ linear? That is, is the length-4 binary repetition code linear?

Ans: Yes, any $\vec{c}_i + \vec{c}_j \in \mathcal{C} \ \forall i, j$.

(eg2) Is $\mathcal{C} = \{(00100), (10010), (01001), (11111)\}$ linear?

Ans: No, $\vec{c}_0 = (00000)$ not a codeword!

② If $\mathcal{C}$ is linear, then d_{min} = least weight non-zero codeword.

Pf: $d_{min} = \min\{d(\vec{c}_i, \vec{c}_j)\} \ \forall i \neq j = \min\{w(\vec{c}_i - \vec{c}_j)\} = \min\{w(\vec{c}_k)\}$; since code is linear, $\vec{c}_k \in \mathcal{C}$.

(eg3) Consider $\mathcal{C} = \{(00000), (00111), (01010), (01101)\}$

Q1: Is this code linear? Ans: Yes, $\vec{c}_i + \vec{c}_j \in \mathcal{C} \ \forall i, j$

Q2: What's d_{min}? Ans: since $\min\{w(\vec{c}_k)\} = 2$, $d_{min} = 2$.

Q3: How many errors is this code guaranteed to correct/detect?

Ans: $t_{detect} \leq d_{min} - 1 = 1$ $\quad t_{correct} \leq \left\lfloor \frac{d_{min}-1}{2} \right\rfloor = 0$

(3) For any linear code with Hamming distance at least $2t+1$,

$$2^{n-k} \geq 1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t}$$

Pf: HD at least $2t+1$ (i.e. $d_{min} = 2t+1$)

$\Rightarrow$ code can correct t or fewer errors!

Number of situations of no error $= 1$

✓ ✓ 1-error $= \binom{n}{1}$

✓ ✓ 2-errors $= \binom{n}{2}$

⋮ ⋮ ⋮

~~t-errors~~

Number of situations of t-errors $= \binom{n}{t}$

So that the total number of these situations is

$1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t}$ — This code can distinguish all these situations.

On the other hand,

There are 2^{n-k} possible distinct parity-bit combinations;

i.e. this code can distinguish at most 2^{n-k} situations.

So that $2^{n-k} \geq 1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t}$ as stated above.

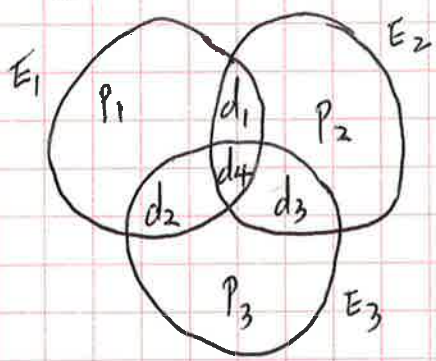
Hamming Codes

Consider the (7, 4) HC discussed in class, with the ff. parity equations:

$$\left.\begin{aligned} P_1 &= d_1 + d_2 + d_4 \\ P_2 &= d_1 + d_3 + d_4 \\ P_3 &= d_2 + d_3 + d_4 \end{aligned}\right\} \text{eqn set } ①$$

where all additions are in GF(2).

It can be seen that $d_{min} = 3$, as the parity calculations contribute at least a weight-2 word. Venn diagrams can be sketched to show which data bits are protected by which parity bit, as below.



The syndrome equations (showing possible error positions) are then given by (from eqn set ①)

$$\left.\begin{aligned} E_1 &= d_1 + d_2 + d_4 + P_1 \\ E_2 &= d_1 + d_3 + d_4 + P_2 \\ E_3 &= d_2 + d_3 + d_4 + P_3 \end{aligned}\right\} \text{eqn set } ②$$

(eg) If $E_3 E_2 E_1 = 101$, only d_2 in error!

(eg) If $E_3 E_2 E_1 = 001$, only P_1 in error!

A syndrome table facilitates these diagnoses!