Note for lecture 4

The error correction/detection techniques covered here are based on the following book:

- Joseph A. Gallian. Contemporary Abstract Algebra. 7th edition. Brooks/Cole, 2010. ISBN 9780495831532 Chapter 31.

Note that in the following, we use the same definition of a "code word" as defined in lecture 3.
Same applies for the definition of a "linear code" (i.e., a parity check code)

Definition 1. Hamming distance between two code words is the # of different bits between the two and is denoted by $d(u,v)$ for code words u and v. (This is essentially the minimum distance we defined in lecture 3)

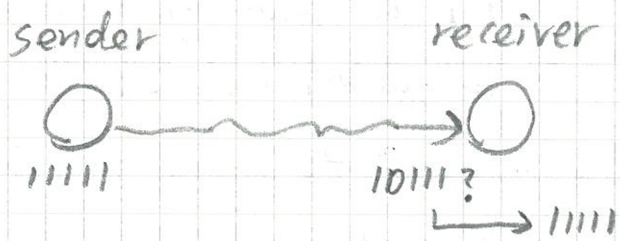
Definition 2. Hamming weight of a code word is the # of non-zero bits in it, denoted by $wt(u)$ for code word u.
"Hamming weight for a linear code" is the minimum Hamming weight of any non-zero code word in the linear code.

Example: let $u=\{00011\}$, $v=\{00010\}$, $w=\{00000\}$
then $d(u,v)=1$, $d(u,w)=2$, $wt(u)=2$
The Hamming weight of $\{u,v,w\}$ is 1.

Definition 3. The nearest-neighbor rule.

In this rule, error correction is performed by converting the received ~~code word~~ bit vector into the code word that has the smallest Hamming distance to the received ~~code word~~ bit vector.

Example: Suppose there are two code words, 11111 and 11001, and that data sender sent 11111 but data receiver got 10111 due to some channel distortion. Using the nearest-neighbor rule, the data receiver can convert 10111 to 11111 since $d(11111, 10111)=1$ and $d(11001, 10111)=3$, and that corrects the error.



Question: is there any performance guarantee of using the nearest-neighbor rule to correct errors?

Answer: Yes see ↓

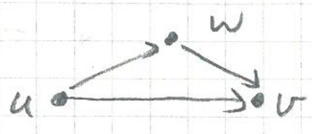
Theorem 1. $d(u,v) = wt(u-v)$ for code words u and v.

Proof idea: In modulo-2 substraction of u by v, the result is a code word having 1s for the bits where u and v differs and 0s for the bits where u and v agrees.

Theorem 2. For any code words u, v, and w,

$$d(u,v) \le d(u,w) + d(w,v).$$

Proof idea:



Theorem 3 (main result!). If the Hamming weight of a linear code is at least $2t+1$, then the nearest-neighbor rule can correct any t or fewer errors; alternatively, it can detect any $2t$ or fewer errors.

Proof idea: Suppose the original code word is u, and the received version is v, and w is any code word other than u.

Then since

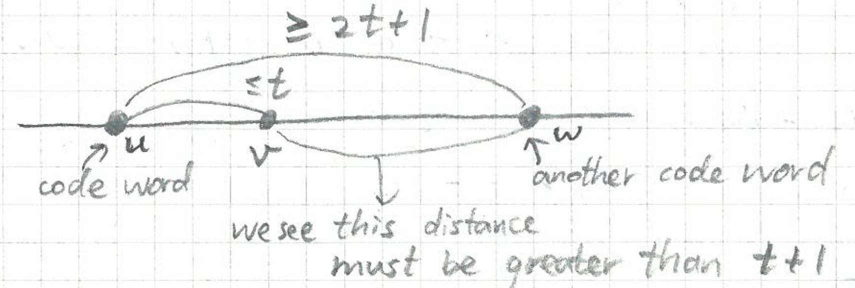
$$\begin{aligned} 2t+1 &\le wt(w-u) = d(w,u) \\ &\le d(w,v) + d(v,u) \\ &\le d(w,v) + t \end{aligned}$$

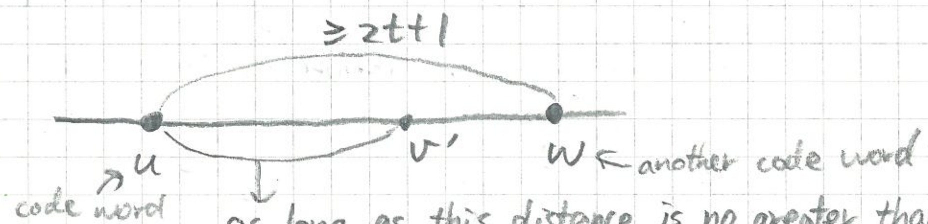
which implies $d(w,v) \ge t+1$.

By definition we have $d(u,v) \le t$.

Therefore u is the closest code word to v, and thus using the nearest-neighbor rule in this case we can successfully correct the error.

Geometric illustration:

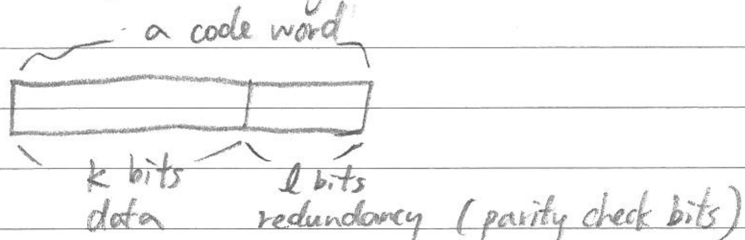


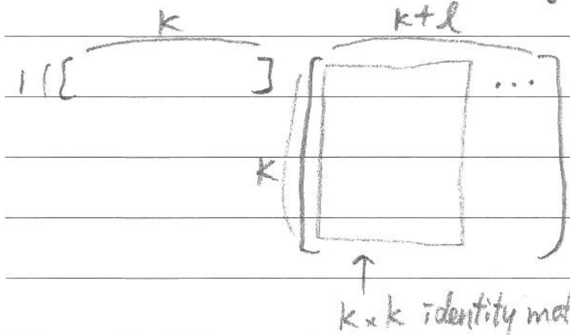


as long as this distance is no greater than $2t$, we can assure that the received bit vector ~~code~~ v' can never be mistakenly identified as a code word; in other word, the error can always be detected (since it is not a code word).

Lecture note 2020/10/05

- Background of Hamming code:





this is called Standard Generator Matrix (a.k.a. Standard Encoding Matrix)

Example Standard Generator Matrix:

$$\begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&1 \\ 0&0&0&1&0&1&1 \end{bmatrix}$$ ← a linear code using this one has Hamming weight = 3

0000 → 0000000	0110 → 0110010
0001 → 0001011	0101 → 0101110
0010 → 0010111	0011 → 0011100
0100 → 0100101	1110 → 1110100
1000 → 1000110	1101 → 1101000
1100 → 1100011	1011 → 1011010
1010 → 1010001	0111 → 0111001
1001 → 1001101	1111 → 1111111

which means, when using the ~~nearest-neighbor rule~~, it can correct any 1-bit error; alternatively, it can detect any 2 or fewer errors.

- Derivation of $E[X] = \sum_{k=1}^{\infty} k\, q(1-q)^{k-1} = q^{-1}$:

Helper lemma 1: $\frac{df(x)}{dx} + \frac{dg(x)}{dx} = \frac{d(f(x)+g(x))}{dx}$

Helper lemma 2: $\frac{d\, f(x)g(x)}{dx} = \frac{df(x)}{dx}\cdot g(x) + f(x)\cdot\frac{dg(x)}{dx}$

Now, $\sum_{k=1}^{\infty} k\, q(1-q)^{k-1} = q\sum_{k=1}^{\infty} k(1-q)^{k-1}$

let $n = 1-q$ ⟶ $= q\sum_{k=1}^{\infty} k\, n^{k-1}$

$= q\sum_{k=1}^{\infty} \frac{d(n^k)}{dn}$

using lemma 1 ⟶ $= q\,\frac{d}{dn}\sum_{k=1}^{\infty} n^k$ Geometric series

$= q\,\frac{d}{dn}\cdot\frac{n}{1-n}$

using lemma 2 ⟶ $= q\left((1-n)^{-1} + n\cdot(-1)(1-n)^{-2}(-1)\right)$

$= q\left(\frac{1}{1-n} + \frac{n}{(1-n)^2}\right)$

$= q\cdot\frac{1}{(1-n)^2}$

since $n = 1-q$ ⟶ $= q\cdot\frac{1}{(1-(1-q))^2}$

$= q^{-1}$

※