MA294 Lecture

Timothy Kohl

Boston University

July 21, 2021

Correcting Errors

The scheme we will explore is the nearest neighbor decoding principle, namely that if an erroneous codeword was transmitted, then we can assume that the nearest codeword in C is the correct one.

Theorem

A code C will correct 'e' errors by the nearest neighbor principle provided that the minimum distance δ satisfies the inequality

$$\delta \geq 2e + 1$$

.

The proof of this isn't too difficult, and makes nice use of the triangle inequality we saw earlier.

Proof.

Say $\vec{c}$ was sent and e errors were made, yielding the bit vector $\vec{z}$.

As such we have $\partial(\vec{c},\vec{z}) = e$.

If now $\vec{c}^{\,'}$ is any other valid codeword then

$$\partial(\vec{c},\vec{z}) + \partial(\vec{z},\vec{c}^{\,'}) \geq \partial(\vec{c},\vec{c}^{\,'}) \geq \delta \geq 2e+1$$

Thus $e + \partial(\vec{z},\vec{c}^{\,'}) \geq 2e+1$ and so $\partial(\vec{z},\vec{c}^{\,'}) \geq e+1$ which implies that $\vec{c}$ is the nearest codeword to $\vec{z}$ and so by the nearest neighbor principle, we infer that the correct codeword *is* $\vec{c}$. $\square$

Going back to the example:

$$C_3 = \{000000, 111000, 001110, 110011\}$$

since $\delta = 3$ then it can detect 2 (i.e. $\delta - 1$) errors and correct $e = 1$ errors, since $\delta = 2(1) + 1$.

Linear Codes

While codes can be constructed according to the $\delta \geq 2e+1$ criterion, it is not so simple to do.

However, if we use the fact that $V = \mathbb{F}_2^n = \mathbb{F}_2 \times \mathbb{F}_2 \times \cdots \times \mathbb{F}_2$ is a group, then we can utilize all that we know about groups to construct codes whose properties are much easier to determine.

Definition

A code $C \subseteq \mathbb{F}_2^n$ is a linear if whenever $\vec{x}, \vec{y} \in C$ so too is $\vec{x} + \vec{y} \in C$. That is, C is actually a sub-group of V.

By Lagrange's theorem, if $C \leq \mathbb{F}_2^n$ is linear then $|C| \big| |\mathbb{F}_2^n| = 2^n$, and so $|C| = 2^k$ for some $k \leq n$.

We refer to k as the *dimension* of C, which is consistent with the notion of dimension from linear algebra since C is k-dimensional subspace of $V = \mathbb{F}_2^n$.

So for a linear code C, we want to find the relationship between n, k and δ.

We have the following somewhat technical result called the 'Sphere Packing Bound'.

Theorem

If C is a linear code of some length n and dimension k then if e is the maximum number of errors which C will correct then

$$2^{n-k} \geq \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e}$$

where $\binom{n}{r} = \frac{n!}{r!(n-r)!}$.

Proof:

If $\vec{c}$ has length n then there are $\binom{n}{r}$ ways of modifying r bits in $\vec{c}$.

Let $S_e(\vec{c})$ be the set of code words which can be obtained from $\vec{c}$ by altering at most e bits.

We have

$$|S_e(\vec{c})| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e}$$

where $\binom{n}{0} = 1$.

So if C corrects e errors then for $\vec{c},\vec{c}^{'}$ distinct codewords in C we have

$$S_e(\vec{c}) \cap S_e(\vec{c}^{'}) = \emptyset$$

So $V = \mathbb{F}_2^n$ contains $|C| = 2^k$ mutually disjoint subsets of size $|S_e(\vec{c})|$ which implies

$$2^n \geq 2^k \times |S_e(\vec{c})|$$

which implies $2^{n-k} \geq |S_e(\vec{c})| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{e}$ □

For example, say $n = 6$, $k = 3$ and $e = 2$ then we must have

$$\begin{aligned} 2^{n-k} = 2^3 &\geq \binom{6}{0} + \binom{6}{1} + \binom{6}{2} \\ &= 1 + 6 + 15 = 22 \end{aligned}$$

which therefore is impossible.

Suppose instead $n = 6$, $k = 3$ and $e = 1$ then

$$2^{n-k} = 2^3 \geq \binom{6}{0} + \binom{6}{1}$$
$$= 1 + 6 = 7$$

so it's not ruled out!

But this is not a guarantee that C *would* correct $e = 1$ errors.

We need more facts about Linear codes to pin this down.

FACT: If C is a linear code then for $\vec{a}$,$\vec{x}$,$\vec{y} \in C$

$$\partial(\vec{x}+\vec{a},\vec{y}+\vec{a}) = \partial(\vec{x},\vec{y})$$

since both $\vec{x}$ and $\vec{y}$ are altered in the same positions by the addition of $\vec{a}$.

With this in mind, we have the following definition.

Definition

If C is a linear code then the weight is defined by

$$w(\vec{x}) = \partial(\vec{x}, \vec{0})$$

for $\vec{x} \in C$ where $\vec{0}$ is the bit vector of all zeros, i.e. the identity element.

What $w(\vec{x})$ measures then is the number of 1's in $\vec{x}$.

Theorem

Let C be a linear code and let w_{min} be the minimum weight of any codeword in C (except $\vec{0} \in C$) then $\delta = w_{min}$.

i.e. The minimum distance is the minimum weight of all the non-zero codewords in C.

Proof.

Let $\vec{c}^*$ be a codeword in C where $w(\vec{c}^*) = w_{min}$.
Since $\vec{c}^*$ and $\vec{0}$ are in C, we have

$$\delta \leq \partial(\vec{c}^*, \vec{0}) = w(\vec{c}^*) = w_{min}$$

However, if $\vec{c}_1$, $\vec{c}_2$ are two distinct codewords at minimum distance from each other then $\vec{c}_1 - \vec{c}_2$ is a codeword since C is a group and

$$\delta = \partial(\vec{c}_1, \vec{c}_2) = \partial(\vec{c}_1 - \vec{c}_2, \vec{c}_2 - \vec{c}_2) = \partial(\vec{c}_1 - \vec{c}_2, \vec{0}) = w(\vec{c}_1 - \vec{c}_2) \geq w_{min}$$

and so $\delta = w_{min}$ $\square$

The virtue of this theorem is that it's *way* easier to compute w_{min}!

Construction of Linear Codes

As a linear code C is a subspace of the (finite) vector space $V = \mathbb{F}_2^n$ then, in fact, C must be the null-space of a matrix.

Let H be a binary matrix with n columns and $\vec{x}$ a bit string considered as a column vector.

In particular $\vec{0}$ will be used to denote the column vector $\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

And if $H\vec{a} = \vec{0}$ and $H\vec{b} = \vec{0}$ then $H(\vec{a} + \vec{b}) = H\vec{a} + H\vec{b} = \vec{0} + \vec{0} = \vec{0}$.

That is, the set $C = \{\vec{x} \in \mathbb{F}_2^n \mid H\vec{x} = \vec{0}\}$ is a linear code.

And we call H the parity check matrix or simply check matrix.

Example: Let $H = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ and suppose

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

[We use the letter 'H' for Hamming, which are the class of codes we wish to explore.]

Then solving this yields the system

$$\begin{aligned} x_1 + x_3 &= 0 \\ x_2 + x_3 + x_4 &= 0 \end{aligned}$$

and so $x_1 = -x_3$, $x_2 = -x_3 - x_4$ where $\{x_3, x_4\}$ are free variables

$$\begin{aligned} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} &= x_3 \begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \\ &= x_3 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} + x_4 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

where $x_3, x_4 \in \mathbb{F}_2$, which yields the code $C = \{0000, 1110, 0101, 1011\}$.

In general, if C is to be a subspace of $\mathbb{F}_2^n$ then we will assume that H has the following form.

$$H = \begin{pmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1,n-r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2,n-r} \\ & & \ddots & & \vdots & & & \\ 0 & 0 & \dots & 1 & b_{r1} & b_{r2} & \dots & b_{r,n-r} \end{pmatrix} = (I_r|B)$$

where I_r is the $r \times r$ identity matrix, and B is an $r \times n-r$ matrix, and overall H is $r \times n$.

We note that the columns could be rearranged which would result in codewords with the bit order re-arranged.

For $H = \begin{pmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1,n-r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2,n-r} \\ & & \ddots & & \vdots & & & \\ 0 & 0 & \dots & 1 & b_{r1} & b_{r2} & \dots & b_{r,n-r} \end{pmatrix}$ as given then for

$\vec{x} \in \mathbb{F}_2^n$ it acts on, we have $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_r \\ x_{r+1} \\ \vdots \\ x_n \end{pmatrix}$ where $x_{r+1}, \dots, x_n$ will be the free variables in the solution of the homogeneous system $H\vec{x} = \vec{0}$ since the matrix H is already in row reduced echelon form.

The resulting code C will be such that $dim(C) = k = n - r$ since there will be 2^{n-r} choices for $x_{r+1}, \dots, x_n$.

Error Correcting in Linear Codes

Theorem

If no column of H consists entirely of zeros, and no two columns of H are the same, then the code C deriving from the solutions of $H\vec{x} = \vec{0}$ will correct one error.

Basically, what the restrictions on

$$H = \begin{pmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1,n-r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2,n-r} \\ & & \ddots & & \vdots & & & \\ 0 & 0 & \dots & 1 & b_{r1} & b_{r2} & \dots & b_{r,n-r} \end{pmatrix}$$

yield is that, since $H : \mathbb{F}_2^n \to \mathbb{F}_2^r$, one has that $rank(H) = r$, i.e. is as large as possible, which guarantees that $\delta = w_{min} \geq 3$.

For each given r there is a standard class of matrices H which have the property mentioned in the theorem and, in fact, the resulting codes have $\delta = 3$, and these are called the Hamming Codes.

Given r let $n = 2^r - 1$ and $k = 2^r - 1 - r$ and define H as follows:

$$H = \begin{pmatrix} 0 & 0 & \dots & 1 \\ \vdots & \vdots & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \vdots & 1 \\ 1 & 0 & \dots & 1 \end{pmatrix}_{r \times n}$$

where the first column is the binary representation of 1, (i.e. $00\dots01$), the second is the binary representation of 2 (i.e. $00\dots10$) and so on until the n-th column (where $n = 2^r - 1$), that is the bit string $11\dots1$ of length r. And one can see that this matrix has no columns of zeros, and no duplicate columns.

Example: Let $r = 3$ which implies $n = 2^3 - 1 = 7$ so that $k = 2^3 - 1 - 3 = 4$ which yields the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

where we observe that the columns correspond to the binary representations of the numbers $\{1, \ldots, 7\}$ and this code is called, not surprisingly, the Hamming(7,4) code.

We can write out the codewords explicitly.

The matrix $H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$ row reduces (by a simple swap of rows 1 and 3) to $H' = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$ which yields the system of equations:

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7 \\ x_2 &= x_3 + x_6 + x_7 \\ x_4 &= x_5 + x_6 + x_7 \end{aligned}$$

where $x_3, x_5, x_6, x_7 \in \mathbb{F}_2$ are free, yielding 2^4 codewords of length 7.

Hamming(7,4) (built from the row-reduced version H' of H)

$$C = \{0000000, 1110000, 1001100, 0111100,\\ 0101010, 1011010, 1100110, 0010110,\\ 1101001, 0011001, 0100101, 1010101,\\ 1000011, 0110011, 0001111, 1111111\}$$

Error Correction?
If $\vec{c}$ is a codeword in C and is offset by an error $\vec{e}_i$ (i.e an error in bit i) then if $\vec{z} = \vec{c} + \vec{e}_i$ we have

$$H'\vec{z} = H'(\vec{c} + \vec{e}_i) = H'\vec{c} + H'\vec{e}_i = \vec{0} + H'\vec{e}_i$$

where $H'\vec{e}_i$ is the i-th column of H'!

Example:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

where the vector on the right hand side is the 3^{rd} column of H', which means the error is in the third position, so the correct codeword/vector is:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

i.e. 0001111.

Recall that if a code C corrects e errors that

$$|S_e(\vec{c})| = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{e}$$

where $S_e(\vec{c})$ is the number of words that can be made by making at most e errors.

So if C is a code of length n with $\delta = 3$ the number of words that can be made by making at most 1 errors in a given codeword is $S_1(\vec{c})$ where $|S_1(\vec{c})| = \binom{n}{0} + \binom{n}{1} = n + 1$.

Since $\delta = 3$ the $S_1(\vec{c})$ do not overlap so

$$|C| \times (n+1) \leq 2^n$$

and for the Hamming code $|C| = 2^n$ where $k = 2^r - 1 - r$ and $n = 2^r - 1$ so $n + 1 = 2^r$ so $|C| \times (n+1) = 2^k 2^r = 2^n$.

In this situation, the code is said to be perfect.

Cyclic Codes

Definition

A code C is said to be cyclic if it is linear and if

$$a_0a_1\ldots a_{n-1} \in C \to a_{n-1}a_0\ldots a_{n-2} \in C$$

If we view C as a set of codewords represented by bitstrings indexed by $\{0,\ldots,n-1\}$, i.e. $a_0a_1\ldots a_{n-1}$ then permutations $\sigma \in S_n$ can act on these bit strings by permuting the indices.

Namely $\sigma(a_0a_1\ldots a_{n-1}) = a_{\sigma(0)}a_{\sigma(1)}\ldots a_{\sigma(n-1)}$ and for the permutation $\sigma = (0,n-1,n-2,\ldots,1) = (0,1,\ldots,n-1)^{-1}$ we have that $\sigma(a_0a_1\ldots a_{n-1}) = a_{n-1}a_0\ldots a_{n-2}$.

As such, a code C is cyclic if for each $\vec{x} \in C$ one has $\sigma(\vec{x}) \in C$ for $\sigma = (0,n-1,n-2,\ldots,1)$.

For example, the code $C = \{000, 110, 011, 101\}$ is cyclic, where, for $\sigma = (0, 1, 2)$ we have $\sigma(000) = 000$, $\sigma(110) = 011$, $\sigma(011) = 101$ and $\sigma(101) = 110$.

On various CPUs, there are frequently low level instructions such as ROL and ROR (rotate left and rotate right) which implement this kind of cyclical 'rotation' of bits in a 'byte' in a computer.

e.g.

$$\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{1} \overset{ROR}{\rightarrow} \boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}$$

The notation $\vec{a} = a_0 a_1 \dots a_{n-1}$ allows us to relate $\vec{a} \in \mathbb{F}_2^n$ with $a(x) = a_0 + a_1 x + \dots a_{n-1} x^{n-1} \in \mathbb{F}_2[x]$.

If now we shift $\vec{a}$ to get $\hat{a} = a_{n-1} a_0 a_1 \dots a_{n-2}$ we get the corresponding polynomial

$$
\begin{aligned}
\hat{a}(x) &= a_{n-1} + a_0 x + \cdots + a_{n-2} x^{n-1} \\
&= x(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) - a_{n-1}(x^n - 1) \\
&= x(a(x)) - a_{n-1}(x^n - 1)
\end{aligned}
$$

So that, by taking remainders we get

$$\hat{a}(x) \equiv x(a(x)) \ (mod\ x^n - 1)$$

Earlier, we saw that taking remainders of polynomials in $\mathbb{F}_p[x]$ by some irreducible $k(x)$ yielded a finite field.

Here however, $x^n - 1$ is generally not irreducible but the set of remainders mod $x^n - 1$ does still give one a ring.

$$V[x] = \mathbb{F}_2[x]/\langle x^n - 1\rangle = \{b_0 + b_1 x + \cdots + b_{n-1}x^{n-1} \mid b_i \in \mathbb{F}_2\}$$

where addition is the usual addition of polynomials, and multiplication is the usual multiplication of polynomials with the relation $x^n - 1 = 0$, namely $x^n = 1$.

For example: $x^3 - 1$, so $x^3 = 1$, which also implies that $x^4 = x$, $x^5 = x^2$, etc.

$(b_0 + b_1 x + b_2 x^2)(c_0 + c_1 x + c_2 x^2)$

$$= (b_0 c_0) + (b_0 c_1 + b_1 c_0)x + (b_0 c_2 + b_1 c_1 + b_2 c_0)x^2 + (b_1 c_2 + b_2 c_1)x^3 + (b_2 c_2)x^4$$

$$= (b_0 c_0) + (b_0 c_1 + b_1 c_0)x + (b_0 c_2 + b_1 c_1 + b_2 c_0)x^2 + (b_1 c_2 + b_2 c_1) + (b_2 c_2)x$$

$$= (b_0 c_0 + b_1 c_2 + b_2 c_1) + (b_0 c_1 + b_1 c_0 + b_2 c_2)x + (b_0 c_2 + b_1 c_1 + b_2 c_0)x^2$$

So what's the point?

If we look at $a_0 + a_1x + \ldots a_{n-1}x^{n-1} \in V[x]$ and consider the code string, $a_0a_1 \ldots a_{n-1}$ then all such code strings give rise to cyclic codes.

$$i.e.\ \hat{a}(x) \equiv xa(x)$$

namely the shifted/rotated vector $\hat{a}(x)$ is $xa(x)$ which is back in C, in that one can extract the coefficients to get the codeword.

How large is $V[x]$?

Well, look at the remainders:

$$\{a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \mid a_i \in \mathbb{F}_2\}$$

which means there are 2^n choices of the coefficients.

Indeed, $a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ corresponds to $(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_2^n = V$ and we observe that $V = \mathbb{F}_2^n$ is itself linear and closed under cyclic shift.

So... a cyclic code in general corresponds to a particular subset of $V[x]$ just as it corresponds to a particular subset of V itself.
The construction of the particular subset that is cyclic will be discussed in the next lecture.