CS681 **Computational Number Theory**

Lecture 14: BCH Codes

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We shall look a a special form of linear codes called cyclic codes. These have very nice structures underlying them and we shall study BCH codes.

1 General Codes

Recall that a linear code $\mathcal{C}$ is just a subspace of $\mathbb{F}_q^n$. We saw last time that by picking a basis of $\mathcal{C}$ we can construct what is known as a parity check matrix H that is zero precisely at $\mathcal{C}$.

Let us understand how the procedure works. Alice has a message, of length k and she wishes to transmit across the channel. The channel is unreliable and therefore both Alice and Bob first agree on some code $\mathcal{C}$. Now how does Alice convert her message into a code word in $\mathcal{C}$? If Alice's message could be written as $(x_1, x_2, \cdots, x_k)$ where each $x_i \in \mathbb{F}_q$, then Alice simply sends $\sum_{i=1}^{k} x_i b_i$ which is a codeword.

Bob receives some y and he checks if $Hy = 0$. Assuming that they choose a good distance code (the channel cannot alter one code into another), if Bob finds that $Hy = 0$, then he knows that the message he received was untampered with.

But what sort of errors can the channel give? Let us say that the channel can change atmost t positions of the codeword. If x was sent and x' was received with atmost t changes between them, then the vector $e = x' - x$ can be thought of as the error vector. And since we assumed that the channel changed atmost t positions, the error vector can have weight atmost t.

This then means that Alice sent an x and Bob received $x + e$. Bob runs the parity check matrix on $x + e$ to get $H(x + e) = Hx + He = He$. The quantity He is called the *syndrome*, which is just the evaluation of Bob's received message by the parity check matrix. If the syndrome is zero, Bob knows that the received word is a valid codeword.

Of course, in order to determine what the actual message was, Bob needs to figure out what e is (for then he knows the message was $x' - e$) but recovering e from He is still a hard thing. It is not clear how this can be done efficiently on a general setting.

2 Cyclic Codes

Definition 1. *A cyclic code $\mathcal{C}$ is a linear code such that if $(c_0, c_1, \cdots, c_{n-1})$ is a codeword, then so is $(c_{n-1}, c_0, c_1, \cdots, c_{n-2})$. To put it algebraically, the space of codewords is invariant under cyclic shifts.*

If course any codeword that is shifted by i places, to the left or the right, will also be a codeword. In order to be able to see the strong structure behind them, we need a different perspective on $\mathbb{F}_q^n$.

2.1 Codewords as Polynomials

Given a vector $(c_0, c_1, \cdots, c_{n-1})$, we can associate a polynomial naturally which is $c(X) = c_0 + c_1X + \cdots + c_{n-1}X^n$. This is just interpretting the vector space $\mathbb{F}_q^n$ as the additive group of the ring $\mathbb{F}_q[X]/(f(X))$ where f is a polynomial of degree n, since they are both isomorphic.

The ring picture has the extra multiplicative structure which is very useful here. Suppose we have a codeword $c = (c_0, \cdots, c_{n-1})$, what can we say about the codeword $c' = (c_{n-1}, c_0, \cdots, c_{n-2})$? As a polynomial, $c = c_0 + c_1X + \cdots + c_{n-1}X^{n-1}$ and $c' = c_{n-1} + c_0X + \cdots + c_{n-2}X^{n-1}$. So essentially we just took the polynomial c and multiplied by X. The last term $c_{n-1}X^n$, however, was changed to c_{n-1}. How do we achieve this? Do the multiplication modulo $X^n - 1$ which is just identifying X^n by 1.

Thus, cyclic shifts is just multiplication of polynomials in $\mathbb{F}_q[X]/(X^n - 1)$ by powers of X. With this observation, the following theorem summarizes the strong underlying structure in cyclic codes.

Theorem 1. *Any cyclic code $\mathcal{C}$ is an ideal of $R = \mathbb{F}_q[X]/(X^n - 1)$. And conversely, every ideal is a cyclic code*

Proof. Let us prove the easier converse first. Let $f(X) \in R$ be an element of the ideal $\mathcal{C}$. Then it follows that for any polynomial $a(X)$, $a(X)f(X) \in \mathcal{C}$ and in particular $X^i f(X) \in \mathcal{C}$. But we already say that multiplying by powers of X was just shifting and therefore our code is also invariant under shifts.

The other direction is straightforward too. We want to show that given a cyclic code $\mathcal{C}$, for any code word $f(X)$ and any polynomial $a(X)$, $a(X)f(X) \in \mathcal{C}$.

$$\begin{aligned} a(X)f(X) &= (a_0 + a_1X + \cdots + a_{n-1}X^{n-1})f(X) \\ &= a_0f(X) + a_1(Xf(X)) + \cdots + a_{n-1}(X^{n-1}f(X)) \\ &= a_0f_0(X) + a_1f_1(X) + \cdots + a_{n-1}f_{n-1}X \qquad X^if(X) \text{ is shifting} \\ &= f'(X) \in \mathcal{C} \end{aligned}$$

□

Suppose $X^n - 1$ factorizes into irreducible polynomials over $\mathbb{F}_q$, say

$$X^n - 1 = g_1g_2 \cdots g_k$$

Then it easy to check that infact all ideals of R are principle, of the form $g(X)R$ where $g(X)$ is a factor of $X^n - 1$. And hence, we have a simple corollary to above theorem.

Corollary 2. *Every cyclic code $\mathcal{C}$ is just the set of multiples of a single polynomial $g(X) \in R$.*

This polynomial is called the generator polynomial. Let us say we pick a factor $g(X)$ of $X^n - 1$ and let its degree be d. What can we say about the dimension of the code $(g(X))$? For this, we will need the rank-nullity theorem.

Theorem 3 (Rank-Nullity)**.** *If T is a linear map from a between two vector spaces V and W, then rank(T) + nullity$(T) = \dim V$ where rank(T) is defined to be the dimension of the image of V and nullity the dimension of the kernel.*

Now look at the map $\phi : R \longrightarrow R/(g(X))$. This, being a homomorphism of rings will also be a linear map on the additive groups which are vector spaces. The dimension of R is n and the dimension of the image, which is $R/(g(X))$, is d. And therefore, the dimension of the kernel which is $\mathcal{C} = (g(X))$ is $n - d$.

What about the parity check matrix? That is extremely simple here. Since the ideal is generated by a single polynomial $g(X)$, we just need to check if any given polynomial is in the code or not by just checking if g divides it. Thus, just the modulo operation is the parity check. This can be written as a matrix as well but the idea is clear.

3 BCH Codes

BCH[1] codes is an example of a cyclic code that is widely studied in coding theory. In order to get a cyclic code, we just need to get the generating polynomial of that code.

Instead of asking for the polynomial in terms of the coefficient, what if we identify the polynomial by the roots instead? This is the general idea of a BCH code.

We are working in a vector space of dimension n over $\mathbb{F}_q$ and identifying cyclic codes as ideals of $R = \mathbb{F}_q[X]/(X^n - 1)$. Let us further impose the constraint that the roots of $X^n - 1$ are distinct by making sure $gcd(n, q) = 1$ so that the derivative is non-zero.

Let ζ be a primitive n-th root of unity in R and look at the set $\{\zeta, \zeta^2, \cdots, \zeta^d\}$ where $d < \phi(n)$ (to prevent some $\zeta^i = \zeta^j$)[2]. Now we ask for the smallest degree polynomial g that has ζ^i as a root for $1 \leq i \leq d$. This polynomial is going to be our generating polynomial for the cyclic code.

The parity check matrix of a BCH code is pretty simple. Note that if $c(X) \in \mathcal{C}$, then $c(X)$ is a multiple of $g(X)$ and in particular $c(X)$ will also have the ζ^i as roots. And therefore, all we need to check is if $c(\zeta^i) = 0$ for all $1 \leq i \leq d$. Now interpretting $c(X)$ as a vector $(c_0, c_1, \cdots, c_{n-1})$ of coefficients, the parity check reduces to the following matrix multiplication.

$$\begin{bmatrix} 1 & \zeta & \zeta^2 & \cdots & \zeta^{n-1} \\ 1 & \zeta^2 & (\zeta^2)^2 & \cdots & \zeta^{2^{n-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^d & (\zeta^d)^2 & \cdots & (\zeta^d)^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note that the parity check matrix H is a $(n - d) \times d$ matrix.

3.1 Distance of a BCH Code

Suppose $g(X)$ is the generating polynomial for the set being the first d powers of ζ, what can we say about the distance of the cyclic code $(g(X))$?

Theorem 4. *A BCH code obtained by considering the first d powers of ζ has distance $d + 1$.*

[1] Bose, Ray-Chaudhuri, Hocquenghem

[2] why won't $d < n$ suffice?

Proof. We would like to show that the minimum weight of the code $\mathcal{C} = (g(X))$ has to be atleast $d+1$. Suppose not, then there is a codeword c such that the weight of c is less than or equal to d. Then this polynomial has atmost d positions with non-zero entries. Let us denote those coefficients by $\{c_{i_1}, c_{i_2}, \cdots, c_{i_d}\}$ and say in increasing order of indices.

We just need to check that for each $1 \leq k \leq d$

$$\sum_{j=1}^{d} c_{i_j} (\zeta^k)^{i_j} = 0$$

But the above equation corresponds to the following matrix product

$$\begin{bmatrix} \zeta^{i_1} & \zeta^{i_2} & \cdots & \zeta^{i_d} \\ (\zeta^{i_1})^2 & (\zeta^{i_2})^2 & \cdots & (\zeta^{i_d})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\zeta^{i_1})^d & (\zeta^{i_2})^d & \cdots & (\zeta^{i_d})^d \end{bmatrix} \begin{bmatrix} c_{i_1} \\ c_{i_2} \\ \vdots \\ c_{i_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note that the $d \times d$ matrix is essentially in the form of a vandermonde matrix:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}$$

and it is well known that the determinant of this matrix is $\prod_{i<j}(x_i - x_j)$ and therefore non-zero if each x_i is distinct as in our case of ζ^{i_j}. Therefore, $Hc = 0$ and H being invertible forces that c has to be the zero vector as well!

Therefore, the only codeword that can have weight less than or equal to d is the zero vector. And therefore the minweight of the BCH code is atleast $d+1$. □

Now that we have this, we can use $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ to get a guarantee that our code has distance atleast d. This is called the *designed distance* of the BCH code. Note that the actual distance you could be larger than d. We just have a guarantee that it is atleast d but the could potentially give you codes of larger distance. There are examples of BCH codes with the actual distance larger than the designed distance.