CS681 **Computational Number Theory**

Lecture 15 and 16: BCH Codes: Error Correction

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

In these two lectures, we shall see how error correction is done in a BCH code.

(most of the class was spent on discussing the solutions for the mid-semester examination)

1 Error Correction in a BCH Code

Recall that a cyclic code is one where the space of codewords is also invariant under cyclic shifts. Last class we identified this with ideals of the ring $\mathbb{F}_q[X]/(X^n - 1)$. We also said that this is an principle ideal domain when $gcd(n, q) = 1$ and therefore every cyclic code can be identified by the polynomial that generates the ideal.

For the BCH code, we pick a principle root of unity ζ and the generator of code is chosen to be the least degree polynomial that has $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ and we argued that the distance of that cyclic code is guaranteed to be at lease d.

How about decoding? Suppose Alice sent a message and that was corrupted at atmost $\lfloor \frac{d}{2} \rfloor$ places, can Bob recover the message efficiently? The answer is yes, and we shall see how. Most of the details shall be done in the next class.

1.1 The Locator and Correction Polynomials

Alice is going to send some polynomial whose degree is bounded by $n-1$, say $c(X) = c_0 + c_1X + \cdots + c_{n-1}X^{n-1}$ and Bob would receive $c(X) + e(X)$ where e is the error. Suppose the channel can corrupt at most $\lfloor \frac{d-1}{2} \rfloor$ coefficients, we know that the number of non-zero coefficients of $e(X)$ is less than $d/2$. Let $M = \{i \ : \ e_i \neq 0\}$. And hence, $|M| = t \leq \frac{d-1}{2}$.

Now look at the two polynomials:

$$\begin{aligned} u(Y) &= \prod_{i \in M} (1 - \zeta^i Y) \\ v(Y) &= \sum_{i \in M} e_i \zeta^i Y \prod_{i \neq j \in M} (1 - \zeta^j Y) \end{aligned}$$

The polynomial $u(X)$ is called the *locator polynomial* and $v(Y)$ is called the *correction polynomial.* Suppose we have $u(Y)$, how do we find out which places of the message are corrupted? This is clear because $u(\zeta^{-i}) = 0$ if and only if $i \in M$ and therefore by just checking $u(\zeta^{-i})$ for all i, we would know precisely at what places the corruption happened.

OK, now we know where the corruption has happened. How do we find out what that coefficient of $e(Y)$ was so that we can recover the message? This is where $v(Y)$ comes in. Notice that v isn't too different from the formal derivative of u. By the chain rule, we can show that

$$u'(Y) = -\sum_{i \in M} \zeta^i \prod_{i \neq j \in M} (1 - \zeta^j Y)$$

So first find out, using the detection polynomial, the places at which the error has occured. Suppose i was one of the places, what can we say about $v(\zeta^{-i})$? Note that every term in the sum other than i will be killed as there would be the term $(1 - \zeta^i Y)$ in the product that is zero when $Y = \zeta^{-i}$. And therefore,

$$\begin{aligned} v(\zeta^{-i}) &= e_i \zeta^i \zeta^{-i} \prod_{i \neq j \in M} (1 - \zeta^j \zeta^{-i}) \\ &= e_i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \end{aligned}$$

What about $u'(\zeta^{-i})$? For the same reason, the only surviving term in the summation would be the one with i. Therefore,

$$\begin{aligned} v(\zeta^{-i}) &= e_i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \\ u'(\zeta^{-i}) &= -\zeta^i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \\ \implies \frac{v(\zeta^{-i})}{u'(\zeta^{-i})} &= -\frac{e_i}{\zeta^i} \end{aligned}$$

And we are done! Running over all detected places, we can completely recover the polynomial $e(X)$ and therefore the actual message.

All that's left to do is find out how to compute the polynomials $u(Y)$ and $v(Y)$. Once we do that, we can locate the errors and also correct them. Finding these two polynomials is the key.

1.2 Computing the Polynomials

There are two important things to note here.

- We don't need to find u and v exactly. Any αu and αv, where α is some constant, would do. We just want u and v up to a scale.
- The degree of u and v is $|M| = t \leq \frac{d-1}{2}$

And remember, all that Bob has is the knowledge that the code is constructed from $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ and the received word $r(X)$. From this, he needs to compute $u(Y)$ and $v(Y)$ to error-correct.

First, let us look at the following rational function

$$w(Y) = \frac{v(Y)}{u(Y)} = \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y}$$

At this point, let us make an outrageous mathematically incorrect Taylor expansion but justify it later in the lecture. Note that we have a term of the form $\frac{1}{1-\zeta^i Y}$ and we are going to expand this as a geometric series.[1]

[1] mathematically minded people are requested to clench their fists and tolerate this for a while. it will be justified soon.

Then, we have

$$\begin{aligned}
w(Y) &= \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y} \\
&= \sum_{i \in M} e_i \zeta^i Y \left(\sum_{k=0}^{\infty} (\zeta^i Y)^k \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} \left(\sum_{i \in M} e_i (\zeta^i)^k \zeta^i \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} \left(\sum_{i \in M} e_i (\zeta^{k+1})^i \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} e(\zeta^{k+1}) \\
&= \sum_{k=1}^{\infty} Y^k e(\zeta^k)
\end{aligned}$$

The first $d-1$ coefficient of $w(Y)$ can be found out easily as we can find $e(\zeta^k)$ easily. Bob has the received code word $r(X) = c(X) + e(X)$. He doesn't know what $e(X)$ or $c(X)$ is but all he needs to do is compute $r(\zeta^k)$. Note that since c is the message, c is a multiple of $g(X)$ and ζ^k is a root of g and hence c. Therefore, $r(\zeta^k) = c(\zeta^k) + e(\zeta^k) = e(\zeta^k)$.

Justifying the Mathematical Sin

Of course, you just cannot write every $\frac{1}{1-x}$ as $1 + x + x^2 + \cdots$. For example, $\frac{1}{1-2}$ is certainly not $1 + 2 + 2^2 + \cdots$. So how do we justify it here?

Now the first thing is that we cannot hope to do anything better than $d-1$ coefficients of $w(Y)$ since we just know that $c(X)$ has $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ as roots. Therefore, we shall focus on finding $w(Y)$ up till the $(d-1)$-th coefficient. By this, we just mean that we are finding $w(Y) \bmod Y^d$, which is just making $Y^d = 0$ in the expression.

Now note that $1 - (\zeta^i Y)^d = 1 - \zeta^{id} Y^d = 1$ and also that $(1-x)$ divides $(1-x^d)$ and hence $(1 - \zeta^i Y)$ divides $(1 - (\zeta^i Y)^d) = 1$. Which means that there exists some polynomial $p(Y)$ such that $(1 - \zeta^i Y) p(Y) = 1$ and hence $(1 - \zeta^i Y)$ is invertible modulo Y^d.

Hence, we can rework as follows:

$$\begin{aligned} w(Y) &= \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y} \bmod Y^d \\ &= \sum_{i \in M} e_i \zeta^i Y \cdot \left(1 - \zeta^i Y\right)^{-1} \bmod Y^d \end{aligned}$$

and it is easy to check that the inverse of $(1-\zeta^i Y)$ modulo Y^d is $\sum_{k=0}^{d-1} (\zeta^i Y)^k$ and hence we have the rest of the equations going through.

$$w(Y) = \sum_{k=1}^{d-1} Y^k e(\zeta^k) \bmod Y^d$$

OK, we now have $w(Y)$, how do we use that to get u and v? The idea is to solve a system of equations to get u and v. Remember that both u and v are degree t polynomials and moreover the constant term in u is 1 and the constant term in v is 0.

Here we shall give an intuitive reasoning and not go into the details of the method. Suppose $u(Y) = 1 + u_1 Y + \cdots + u_t Y^t$ and $v(Y) = v_1 Y + \cdots v_t Y^t$, then we can just think of coefficients as some parameters to evaluate. Using the values of $w(Y) \bmod Y^k$ for all $1 \leq k \leq d$, we can solve for u_i, v_i by writing a system of equations. And since the number of parameters we need to solve for is $2t$ and this is less than or equal to the number of equations we have, it can be done efficiently.

There is infact another approach to solve for u and v using something called the Berlekamp-Welch Decoder. We won't be covering this in the course but just to tell you that the locator and corrector polynomials can be computed efficiently from the received word $r(X)$.

Hence using such efficient algorithms we can compute u and v. Then we just use u to locate the errors and then v to correct them at those places. And from the analysis, it is clear that we can't hope to correct more than t errors as we would then have more parameters than equations and there may not exist a solution to that set of equations.

Thus, a BCH code of designed d can be error corrected if the number of errors is bounded above by $\lfloor \frac{d-1}{2} \rfloor$.