Information and Coding Theory **Autumn 2014**

Lecture 12: November 6, 2013

Lecturer: Madhur Tulsiani Scribe: David Kim

Recall: We were looking at codes of the form $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$, where q is prime, k is the message length, and n is the block length of the code. We also saw C (its range) as a set in $\mathbb{F}_q^n$ and defined the distance of the code as

$$\Delta(C) := \min_{x,y \in C, x \neq y} \{\Delta(x,y)\}$$

where $\Delta(x,y)$ is the Hamming distance between x and y. We showed that a code C can correct t errors iff $\Delta(C) \geq 2t+1$.

1 Hamming Code

The following is an example of the Hamming Code from $\mathbb{F}_2^4$ to $\mathbb{F}_2^7$:

Example 1.1 *Let $C : \mathbb{F}_2^4 \to \mathbb{F}_2^7$, where*

$$C(x_1,x_2,x_3,x_4) = (x_1,x_2,x_3,x_4,x_2+x_3+x_4,x_1+x_3+x_4,x_1+x_2+x_4).$$

Note that each element of the image is a linear function of the x_i's, i.e., one can express C with matrix multiplication as follows:

$$C(x_1,x_2,x_3,x_4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Definition 1.2 (Linear Codes) *A code $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is a linear code if for all $u, v \in \mathbb{F}_q^k$ and $\alpha \in \mathbb{F}_q$, $C(\alpha u + v) = \alpha C(u) + C(v)$, and thus the image of C is a subspace of $\mathbb{F}_q^n$.*

Since a linear code is a linear map from a finite dimensional vector space to another, we can write it as a matrix of finite size. That is, there is a corresponding $G \in \mathbb{F}_q^{n\times k}$ s.t. $C(x) = Gx$ for all $x \in \mathbb{F}_q^k$. If the code has nonzero distance, then the rank of G must be k (otherwise there exist $x, y \in \mathbb{F}_q^k$ such that $Gx = Gy$). Hence, the null space of G^T has dimension $n-k$, so let $b_1, \ldots, b_{n-k}$ be a basis of the null space of G^T.

Definition 1.3 (Parity Check Matrix) *Let $b_1, \ldots, b_{n-k}$ be a basis for the null space of G^T corresponding to a linear code C. Then $H \in \mathbb{F}_q^{(n-k)\times n}$, defined by*

$$H^T = \left[\begin{array}{c|c|c|c} b_1 & b_2 & \ldots & b_{n-k} \end{array}\right]$$

is called the parity check matrix of C.

Since $G^T H^T = 0 \Leftrightarrow HG = 0$, we have $(HG)x = 0$ for all $x \in \mathbb{F}_q^k$, i.e., $Hz = 0$ for all $z \in C$. Moreover, since the columns of H^T are a basis for the null-space of G^T, we have that

$$z \in C \;\Leftrightarrow\; Hz = 0\,.$$

So the parity check matrix gives us a way to quickly check a codeword, by checking the parities of some bits of z (each row of H gives a parity constraint on z). Also, one can equivalently define a linear code by either giving G or the parity check matrix H.

Example 1.4 *The parity check matrix of our example Hamming Code is:*

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Note that the i^{th} column is the integer i in binary. One can easily check that $HG = 0$.

Now suppose $z = (z_1, \ldots, z_7)^T$ is our codeword and we make a single error in the i^{th} entry. Then the output codeword with the error is

$$z + e_i = \begin{bmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_7 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

and $H(z + e_i) = Hz + He_i = He_i = H_i$, the i^{th} column of H, which reads i in binary. So this is a very efficient decoding algorithm just based on parity checking. Since the Hamming code (C) can correct at least $t = 1$ errors, we must have that $\Delta(C) \geq 2t + 1 = 3$. Verify that the distance is exactly 3 using the following characterization of distance for linear codes.

Exercise 1.5 *For $z \in \mathbb{F}_q^n$, let $\mathsf{wt}(z) = |\{i \in [n] \;\mid\; z_i \neq 0\}|$. Prove that for a linear code C*

$$\Delta(C) \;=\; \min_{z \in C} \mathsf{wt}(z)\,.$$

One can generalize the Hamming code to larger message and block lengths, we can create a parity matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, where the i^{th} column reads i in binary.

2 Hamming Bound

We now show an optimality bound on the size of the code, starting with the case of distance-3 codes and then generalizing to distance-d codes.

Theorem 2.1 *Let $C : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be any distance-3 code, i.e., $\Delta(C) = 3$. Then*

$$|C| = 2^k \leq \frac{2^n}{n+1}$$

Proof: For each $z \in C$, let $B(z)$ be the ball of size $n+1$ consisting of z and the n elements in $\mathbb{F}_2^n$ (not in C), each at distance 1 from z. Then the balls formed by the codewords in C are disjoint, if $B(z)$ and $B(z')$ intersect, then $\Delta(z, z') \leq 2$ by triangle inequality. For each codeword $z \in C$, we have $|B(z)| = n + 1$ codes, so $|C|(n+1) \leq 2^n$. ■

Note that our example hamming code from $\mathbb{F}_2^4$ to $\mathbb{F}_2^7$ satisfied $|C| = 2^4 = \dfrac{2^7}{8}$, so it was an optimal distance-3 code. Generalizing to distance-d codes, we have:

Theorem 2.2 (Hamming Bound) *Let $C : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be any distance-d code, i.e., $\Delta(C) = d$. Then*

$$|C| = 2^k \leq \frac{2^n}{vol\left(B_{\lfloor \frac{d}{2} \rfloor}\right)}$$

where $vol\left(B_{\lfloor \frac{d}{2} \rfloor}\right) = \sum_{i=1}^{\lfloor \frac{d}{2} \rfloor} \binom{n}{i}$ is the number of codes at distance at most $\lfloor \frac{d}{2} \rfloor$ from any fixed codeword $z \in C$.

Remark 2.3 *The Hamming bound also gives us a bound on the rate of the code in terms of entropy (recall: the rate of the code is $\frac{k}{n}$). Let $d = \delta n$ for $\delta \leq \frac{1}{2}$. Since $\sum_{i=1}^{l} \binom{n}{i} \leq 2^{nH(\frac{l}{n})}$ for $l \leq \frac{n}{2}$, we have:*

$$\frac{k}{n} \leq 1 - H(\delta/2) + o(1).$$

3 Reed-Solomon Code

We now look at Reed-Solomon codes over $\mathbb{F}_q$. These are optimal codes which can achieve a very large distance. However, they have a drawback that they need $q \geq n$.

Definition 3.1 (Reed-Solomon Code) *Assume $q \geq n$ and fix $S = \{a_1, \ldots, a_n\} \subseteq \mathbb{F}_q$, distinct s.t. $|S| = n$. For each message $(m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k$, consider the polynomial $P(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$. Then the Reed-Solomon Code is defined as:*

$$C(m_0, \ldots, m_{k-1}) = (P(a_1), \ldots, P(a_n)).$$

Remark 3.2 *Reed-Solomon Codes can again be encoded using "inner codes" to create "concatenated codes", which can work with smaller q. However, we will not discuss these.*

Let's compute the distance of the Reed-Solomon Code:

Claim 3.3 $\Delta(C) \geq n - k + 1$.

Proof: Consider $C(m_0,\dots,m_{k-1})$ with $P(x) = m_0+m_1x+\dots+m_{k-1}x^{k-1}$ and $C(m'_0,\dots,m'_{k-1})$ with $P'(x) = m'_0+m'_1x+\dots+m'_{k-1}x^{k-1}$, where we assume $(m_0,\dots,m_{k-1}) \neq (m'_0,\dots,m'_{k-1})$ and hence $P \neq P'$. Then $P - P'$ is a non-zero polynomial of degree at most $k-1$ and has at most $k-1$ roots, i.e., P and P' agree on at most $k-1$ points. This implies that $C(m_0,\dots,m_{k-1}) = (P(a_1),\dots,P(a_n))$ and $C(m'_0,\dots,m'_{k-1}) = (P'(a_1),\dots,P'(a_n))$ differ on at least $n-k+1$ places. Since our choice of the messages was arbitrary, $\Delta(C) \geq n-k+1$. ■

We now show that this is optimal:

Theorem 3.4 (Singleton Bound) *Let $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$ be a distance-d code. Then*

$$d \leq n-k+1.$$

Proof: Consider the map $\Gamma : \mathbb{F}_q^k \to \mathbb{F}_q^{n-d+1}$, where $\Gamma(x)$ is the first $n-d+1$ coordinates of $C(x)$. Then for all $x \neq y$, we have $\Gamma(x) \neq \Gamma(y)$, since $\Delta(C(x),C(y)) \geq d$. Therefore, $|img(\Gamma)| = |\mathbb{F}_q^{n-d+1}| \geq |\mathbb{F}_q^k|$. ■

Remark 3.5 *If $n = 2k$, then for $d = k+1$, we can correct $\lfloor \frac{k}{2} \rfloor$ errors using the Reed-Solomon Code, and this is optimal. Moreover, the Reed-Solomon Code is a linear code:*

$$C(m_0,\dots,m_{k-1}) = \begin{bmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & a_n^2 & \dots & a_n^{k-1} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \end{bmatrix}$$

4 Berlekamp-Welch Decoding Algorithm

If the codewords output by the Reed-Solomon code did not contain any errors, we could simply use Lagrange interpolation to recover the messages. However, we must be able to handle noise, and the trick is to work with a hypothetical *error-locator polynomial* telling us where the error is.

Definition 4.1 (Error-locator Polynomial) *An error-locator polynomial, $E(x)$, is a polynomial which satisfies*

$$\forall i \in [n] \quad E(a_i) = 0 \Leftrightarrow y_i \neq P(a_i)$$

for all $i = 1,\dots,n$, where y_i is the i^{th} element of the output Reed-Solomon Code obtained with noise.

Note that this directly implies that $y_iE(a_i) = P(a_i)E(a_i)$ for all $a_i \in S$. Let's denote Q as $Q(x) := P(x)E(x)$. In the next lecture, we will discuss the following algorithm by Berelekemp and Welch [1], which uses the above intuition to decode a message with at most $\lfloor (n-k+1)/2 \rfloor$ errors.

Algorithm 4.2 *Find Q, E with $deg(E) \leq t$ and $deg(Q) \leq k-1+t$ s.t. $y_iE(a_i) = Q(a_i)$ for all $i = 1,\dots,n$. Output $\dfrac{Q}{E}$.*

References

[1] L.R. Welch and E.R. Berlekamp. Error correction for algebraic block codes, December 30 1986. US Patent 4,633,470.