18.434: Seminar in Theoretical Computer Science **Spring 2016**

February 26: Error correcting codes

Lecturer: Lawrence Wu

Contents

1 The Reed-Solomon Codes

Reed-Solomon codes are an important group of error-correcting codes that were introduced by Irving S. Reed and Gustave Solomon in 1960. They are used in applications such as data storage (CDs, DVDs, Blu-ray, RAID 6), and data transmission and broadcast (DSL, WiMAX, DVB, ATSC). We will see in section 3 that they are optimal in a sense and in section 2 we will study computational properties.

Definition 1.1 *An $[n, k, d]_q$ code maps injectively $\mathbb{F}_q^k$ to $\mathbb{F}_q^n$ with minimum Hamming distance d. A **linear code** is such that this map is linear, of full rank. The image C of the map is the **code space**. In the case of a linear map, C is a dimension k subspace of $\mathbb{F}_q$, which is spanned by any image of a basis of $\mathbb{F}_q$. A **generator matrix** expresses such a span as its row vectors.*

Definition 1.2 (Reed-Solomon Codes) *For $1 \leq k < n, q \geq n$, select a subset of symbols of cardinality n, $S \subseteq \mathbb{F}_q$. We define $Enc : \mathbb{F}_q^k \to \mathbb{F}_q^n$ as following:*

For message $m : (m_0, m_1, \cdots, m_{k-1}) \in \mathbb{F}_q^k$,

$$m \mapsto (P_m(a))_{a \in S}$$

where $P_m(x) \in \mathbb{F}_q[x]$ is $m_0 + m_1 x + \ldots + m_{k-1} x^{k-1}$.

Remark: This is a linear code. We can easily check it by definition. We add two messages m and m', $Enc(m + m') = Enc(m) + Enc(m')$ because its like adding the coefficients of polynomials.

Remark: The generator matrix is a Vandermonde matrix. To see this, plug in a basis of $\mathbb{F}_q^k$ to P_m.

Remark: $d \geq n - (k - 1) = n - k + 1$. To see this, recall that d is the minimum Hamming weight over all codewords, but P_m has at most $k - 1$ zeros. In fact, we can show equality, if we let $P_m = \Pi_{\alpha \in S}(x - \alpha)$

In summary, Reed-Solomon codes have parameters $[n, k, n - k + 1]_q$ where we are free to choose n and k subject to the restriction $q \geq n$. Compare this to the Hamming code $[n, n - \log_2(n+1), 3]_2$ and the Hadamard

code $[n, \log_2 n, n/2]_2$. The Reed-Solomon code can beat these codes in rate and error correction with setting $k = \log_2(n+1)$ and $k = n/2$ respectively, but needs a larger alphabet. An intuition for why a larger alphabet would help is that larger symbols can absorb more errors (e.g. an alphabet size of 1024 would have 10 bit symbols which could absorb 10 consective errors).

2 Decoding algorithms

In this section, we consider only the problem of unique decoding, in which we assume that a received message has less than $d/2$ errors, so that by the Triangle Inequality there is a unique codeword within $d/2$ of the received message, from which the original message can be recovered. See [1] for $O(n^2)$ list decoding of up to $1 - \sqrt{R}$ errors in Reed-Solomon codes.

Here we present the Welch-Berlekamp Algorithm for unique decoding and correction of up to $e < \dfrac{n-k+1}{2}$ errors. We can express the received code word in terms of the polynomial

$$P_m(X) = m_0 + m_1 X + \ldots + m_{k-1} X^{k-1} \tag{1}$$

and the degree e monic *error-locator polynomial*

$$E_m(X) = X^a \Pi_{\alpha_i \in S, y_i \neq P_m(\alpha_i)} (X - \alpha_i) \tag{2}$$

where y_i are the symbols of the received codeword and a is selected to achieve the degree of e. In particular, $P_m(X)$ and $E_m(X)$ are solutions to the following system of n equations in $e + (k-1) < \frac{n-k+1}{2} + k - 1 < n$ unknowns (the unknowns are coefficients of $P(X)$ and $E(X)$):

$$y_i E(\alpha_i) = P(\alpha_i) E(\alpha_i) \forall \alpha_i \in S$$

since the equations where $y_i \neq P(\alpha_i)$ evaluate to $0 = 0$. We suspect that solving this system of equations will give us $P(X)$ as desired, but since we are multiplying $P(X)$ with $E(X)$, we are multiplying some of the unknowns together! This makes the system quadratic, and thus NP-hard. However, we can employ a trick called *linearization*, introducing more unknowns to make a linear system. Let us instead solve for the coefficients of the degree $e + k - 1$ polynomial $N(X) = P(X)E(X)$. Our system is now the linear system

$$y_i E(\alpha_i) = N(\alpha_i) \forall \alpha_i \in S \tag{3}$$

of n equations and $e + (e + k) = 2e + k < (n - k + 1) + k = n + 1$ variables which is solvable in $O(n^3)$ time. A nonzero solution clearly exists as given by $E_m(X)$ and $P_m(X)$, so it remains to show that *any* nonzero solution will recover $P_m(X)$.

Claim 2.1 *Any nonzero solution $E(X), N(X)$ to this system will have $\dfrac{N(X)}{E(X)} = P_m(X)$.*

Proof: Consider

$$R(X) = E(X) E_m(X) P_m(X) - E_m(X) N(X).$$

By 1 and 2, we have

$$\begin{aligned} R(\alpha_i) &= E(X) y_{\alpha_i} E_m(X) - E_m(X) y_{\alpha_i} E(X) \\ &= 0 \end{aligned}$$

for all $\alpha_i \in S$. This means $R(X)$ has at least n zeros, but $\deg R(X) \leq 2e + k - 1 < (n - k + 1) + k - 1 = n$, so $R(X)$ must be the zero polynomial. Thus,

$$\begin{aligned} E(X)E_m(X)P_m(X) - E_m(X)N(X) &= 0 \\ E(X)E_m(X)P_m(X) &= E_m(X)N(X) \\ P_m(X) &= \frac{N(X)}{E(X)} \end{aligned}$$

as desired. ■

Since polynomial long division can be done in $O(n^3)$ time, the whole algorithm can be done in $O(n^3)$ time.

3 The existence of good codes

In this section we will show some bounds and existence results of the parameters of codes. The following bound says that each character of distance costs a parity character.

Theorem 3.1 (Singleton bound) *For a $[n, k, d]_q$ code, $k \leq n - d + 1$.*

Proof: If $k > n - d + 1$ then by Pigeonhole Principle, some two elements in C will have same first $n - d + 1$ coordinates, so Hamming distance $\leq d - 1$; contradiction. ■

Note that this bound is independent of q. We can check it for some of the codes we know, such as Hamming codes, Hadamard codes, and Reed-Solomon codes. Note that all three of these codes have restrictions on q. So we still have a question of whether for *fixed* q there exists a code family which is good for infinitely many n. Here we define what we are looking for:

Definition 3.2 (Code family) *Fixed q, let $C = \{C_n\}$ be a sequence of codes $[n, k(n), d(n)]_q$ with monotonically increasing $n \to \infty$.*

Definition 3.3 (Asymptotic rate)

$$R(C) = \lim_{n \to \infty} \inf \frac{k(n)}{n}$$

Definition 3.4 (Asymptotic relative distance)

$$\delta(C) = \lim_{n \to \infty} \inf \frac{d(n)}{n}$$

For example, we have:

- Hamming Code: $R = 1, \delta = 0$
- Hadamard Code: $R = 0, \delta = \frac{1}{2}$
- Reed-Solomon Code: These are not code families because we cannot increase n to ∞ with fixed q. Otherwise, we could get $R = R_0, \delta = 1 - R_0$ depending on how we choose k.

Now, with one more definition, we prove the main result of this section.

Definition 3.5 (Asymptotically good code family) *Asymptotically good code family is a code family such that*

$$R(C) \geq R_0 > 0$$
$$\delta(C) \geq \delta > 0$$

Theorem 3.6 *Given fixed q, asymptotically good code family exists.*

Proof: There is a natural greedy approach to construct a code of distance at least d. We can randomly add a codeword and erase the part within distance d and repeat adding until we cannot proceed. ■

References

[1] http://www.cse.buffalo.edu/faculty/atri/courses/coding-theory/book/chapters/chap12.pdf