CS 6815 Pseudorandomness and Combinatorial Constructions **Fall 2019**

Lecture 15: October 22

Lecturer: Eshan Chattopadhyay *Scribe: Sloan Nietert*

15.1 Unique decoding of Reed-Solomon codes

Recall that each message of the $[n, k, d]_q$ Reed-Solomon (RS) code corresponds to a polynomial p of degree at most $k-1$ over $\mathbb{F}_q$ ($n \leq q$) and is encoded as the evaluation of p at n distinct points, $(p(\beta_1), \ldots, p(\beta_n)) \in \mathbb{F}_q^n$. Earlier, we proved that the distance of this code satisfies $d = n - k + 1$, achieving the Singleton bound. Today, we will begin by completing our description and analysis of the Welch-Berlekamp unique decoding algorithm for Reed-Solomon.

To start, we have a corrupted word $y \in \mathbb{F}_q^n$ (we will take $q = n$) and view it as a function $f : \mathbb{F}_q \to \mathbb{F}_q$, in that $y = (f(\beta_1), \ldots, f(\beta_n))$. Our promise is that there exists some $p \in \mathrm{Poly}_{\leq k-1}$ such that

$$\Pr_{x \in \mathbb{F}_q} [\, p(x) \neq f(x) \,] = \frac{e}{q} \leq \frac{1}{q} \left\lfloor \frac{d-1}{2} \right\rfloor,$$

where e counts the number of errors, i.e. the cardinality of $T = \{x \in \mathbb{F}_q : f(x) \neq p(x)\}$. Thus, we have $e \leq \lfloor (n-k)/2 \rfloor$. Our goal is to find p in $\mathrm{poly}(n)$ time.

The key idea is to consider an error locating polynomial E of degree e such that $E(x) = 0$ if and only if $f(x) \neq p(x)$. Then, we have that

$$E(x)f(x) = E(x)p(x)$$

for each $x \in \mathbb{F}_q$. Expressing the polynomials as $E(x) = \sum_{i=0}^{e} e_i x^i$ and $p(x) = \sum_{i=0}^{k-1} m_i x^i$, this gives a system of quadratic equations which is NP-hard to solve in general. To make this tractable, we will use a "linearizing trick", solving for

$$N(x) = E(x)p(x),$$

a polynomial of degree at most $e + k - 1$. Now, we can full describe the procedure.

Welch-Berlekamp Algorithm
Step 1: Compute a non-trivial solution to the following homogeneous system of linear equations

$$N(x) = E(x)f(x) \quad \forall x \in \mathbb{F}_q$$
$$N(x) = \sum_{i=0}^{t+k-1} n_i x^i \quad E(x) = \sum_{i=0}^{t} e_i x^i$$

for the smallest t possible, starting at $t = \lfloor (n-k)/2 \rfloor \geq e$.

Step 2: If a solution is found and $E(x)$ divides $N(x)$, return $N(x)/E(x)$. Otherwise, the error is uncorrectable, and the promise that $e \leq \lfloor (n-k)/2 \rfloor$ has been broken.

These linear equations can be solved efficiently, so it just remains to prove correctness.

Claim 15.1 *There exists a valid solution to Step 1.*

Proof: Simply take $E^* = \prod_{\alpha \in T}(x - \alpha)$ and $N^*(x) = E^*(x)p(x)$. This implies that the value of t selected for the solution is at most $\deg(E^*) = e$. ■

Claim 15.2 *If (N_1, E_1) and (N_2, E_2) are two valid outputs from Step 1, then $N_1/E_1 = N_2/E_2$.*

Proof: We know that

$$N_1(x)E_2(x) = f(x)E_1(x)E_2(x) = N_2(x)E_1(x)$$

for each $x \in \mathbb{F}_q$, so $N_1E_2 - N_2E_1$ has $q = n$ roots. Further, this polynomial has degree at most

$$(e + k - 1) + e = 2e + k - 1 \leq n - 1,$$

so it must in fact be the zero polynomial. ■

15.2 List decoding

The motivation for list decoding is to "go beyond $d/2$ errors" and, for any potential message, to provide a reasonably small set of possible codewords which might have produced it.

Definition 15.3 *A code $\mathcal{C} \subset \Sigma^n$ is (ρ, L)-list decodable if, for each $y \in \Sigma^n$,*

$$|\operatorname{Ball}(y, \rho n) \cap \mathcal{C}| \leq L,$$

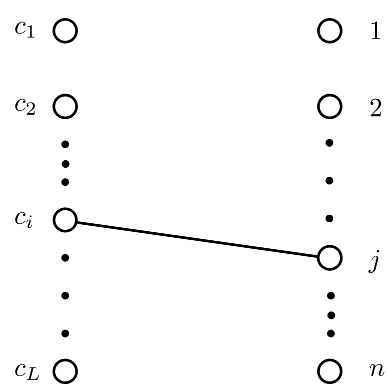
where $\operatorname{Ball}(y, r) := \{w \in \Sigma^n : \Delta(y, w) \leq r\}$.

List decodable codes are useful if $L \leq \operatorname{poly}(n)$ and $\rho > \delta/2$, particularly if $\rho \to \delta - o(1)$. Next, we introduce the Johnson bound, which "translates good distance to good list decodable radius."

Theorem 15.4 (Johnson Bound) *If $\mathcal{C} \subset \mathbb{F}_q^n$ is an error-correcting code with relative distance $\delta(\mathcal{C}) = 1 - \varepsilon$ (i.e. $d = (1 - \varepsilon)n$), then $\mathcal{C}$ is a $(1 - \sqrt{\varepsilon} - o(1), \operatorname{poly}(n))$-list decodable code.*

For the $[n, k, d]_q$ Reed-Solomon code, with $d = n - k + 1$, this translates to a list decodable radius of $1 - \sqrt{(k-1)/n} \sim 1 - \sqrt{r}$. Information theoretic methods give a lower bound of $1 - r - o(1)$, but is an open question whether this can be achieved. Before continuing with the proof of the Johnson bound, we observe two notable drawbacks. First, it is a combinatorial bound that is **not** algorithmic, and, second, it is not tight for all codes.

Proof: Fix $y \in \mathbb{F}_q$, and let $c_1, c_2, \ldots, c_L$ be the codewords in $\operatorname{Ball}(y, \rho n)$. Consider the following graph,



where c_i, j is an edge if and only if $(c_i)_j = y_j$. Observe that (i) the left degree of any c_i is at least $(1-\rho)n$ and that (ii) $|N(c_i) \cap N(c_j)| \le n - d$, where $N(v)$ denotes the neighborhood of vertex v.

Next, we'll consider the expected number of common neighbors between random distinct codewords c_i, c_j. Letting λ_k denote the degree of right vertex k and $\bar{\lambda}$ denote the mean degree of a right vertex, we have

$$n - d \ge \mathbb{E}[\,|N(c_i) \cap N(c_j)|\,] = \frac{\sum_{k=1}^{n} \binom{\lambda_k}{2}}{\binom{L}{2}} \ge \frac{n\binom{\bar{\lambda}}{2}}{\binom{L}{2}},$$

where the second inequality follows from the convexity of the function $x \mapsto \binom{x}{2}$. By double counting, we also know that $\bar{\lambda} \ge (1-\rho)L$, so it follows that

$$\begin{aligned} &(n-d)L(L-1) \ge n\bar{\lambda}(\bar{\lambda}-1) \\ \iff &(n-d)(L-1) \ge (1-\rho)^2 Ln - (1-\rho)n. \end{aligned}$$

After a bit of algebra, we find that

$$L \le \frac{1-\rho}{(1-\rho)^2 - \varepsilon}$$

and choosing $\rho = 1 - \sqrt{\varepsilon} - 1/\operatorname{poly}(n)$ gives the desired $L \le \operatorname{poly}(n)$ bound. ■