Lecture 5: Decoding Reed-Solomon codes

Topics in Error-Correcting Codes (Fall 2022)
University of Toronto
Swastik Kopparty
Scribe: Yibin Zhao and Devansh Shringi

1 Decoding Reed-Solomon Codes

We consider the problem of decoding Reed-Solomon (RS) codes. This is approached by using either the dual view or the primal view of RS codes.

Dual View We first consider decoding Reed-Solomon (RS) codes in the dual view. Given alphabet $\Sigma = \mathbb{F}_q$ for some prime q and string length $n = q$, recall that the parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{t+1} \times \mathbb{F}_q^n$ is defined as

$$\boldsymbol{H} = \begin{bmatrix} & 1 & \\ & \alpha_i & \\ \cdots & \alpha_i^2 & \cdots \\ & \vdots & \\ & \alpha^t & \end{bmatrix}.$$

The Reed-Solomon codes is defined in the dual view as

$$C = \{f \in \mathbb{F}_q^n | \boldsymbol{H} f = 0\}.$$

Definition 1 (Decoding Problem)**.** *Given $r \in \mathbb{F}_q^n$ such that there exists $f \in C$ satisfying $f - r$ has support size at most $\frac{t}{2}$, find f (in* poly(n) *time).*

By applying r on $\boldsymbol{H}$, we get $\boldsymbol{H}r = \boldsymbol{H}f + \boldsymbol{H}e = \boldsymbol{H}e$ where $e = f - r$ is the error vector. This reduces the problem to finding e given $\boldsymbol{H}e = v = (v_i)$. Let $c = |\operatorname{supp}(\boldsymbol{H}e)|$, then the problem of finding e transform into an algebra problem.

$$\begin{aligned} \text{Find} \qquad & \alpha_1, \ldots, \alpha_c \text{ and } \beta_1, \ldots, \beta_c \in \mathbb{F}_q \\ \text{Subject to} \qquad & \sum_{i=1}^{c} \beta_i \alpha_i^k = v_k, \qquad k = 0, \ldots, t \end{aligned} \tag{1}$$

There are $2c$ variables and $t + 1$ equations in this algebraic system. Given that $c \leq \frac{t}{2}$, it has a unique solution. Note that this is not an efficient algorithm due to the system of equations being non-linear.

Algorithm 1 Berlekamp-Welch Algorithm

1: For $c = \frac{n-k}{2} - 1$,

$$\begin{aligned} \text{Find} \quad & e_0, \ldots, e_c \text{ and } b_1, \ldots, b_{k+c} \in \mathbb{F}_q \text{ nontrivial} \\ \text{Subject to} \quad & E(x)r(x) = B(x) \qquad \forall x \in S \\ & \text{where } E(x) = \sum_{i=0}^{c} e_i x^i, B(x) = \sum_{i=0}^{c} b_i x^i. \end{aligned} \tag{3}$$

2: Output $\frac{B(x)}{E(x)}$ if $E(x)|B(x)$. Otherwise, output ERROR.

Primal View Now, consider decoding RS code in the primal view. While many likes the dual view of RS codes for decoding, the primal view is arguably the better view in this problem and many advancements in decoding RS codes specifically drop the dual view.

Given $S \subseteq \mathbb{F}_q$ and $k \leq |S| = n$, recall that the RS code is defined as

$$C = \{f : S \mapsto \mathbb{F}_q | f \text{ is a polynomial of degree} \leq k\}.$$

Definition 2 (Decoding Problem)**.** *Given $r : S \mapsto \mathbb{F}_q$ such that there exists $f \in C$ satisfying f agrees with r on all but $\frac{n-k}{2}$ coordinates, find f (in* poly(n) *time).*

The Berlekamp-Welch algorithm (Algorithm 1) is an efficient decoding algorithm based on the primal view of RS codes. Let $U = \text{supp}(f - r)$. We define $E(x) = \prod_{u \in U}(x - u)$. The main identity that facilitates the algorithm is

$$\forall x \in S, E(x) \cdot r(x) = E(x) \cdot f(x) \tag{2}$$

Notice that the problem in step 1 (eq. (3)) is a linear system with $k + 2c$ variables and n equations. Given $c = \frac{n-k}{2} - 1$, the number of variables is less than n and that the linear system guarantees to have a nontrivial solution and can be solved efficiently.

Consider the correctness of the algorithm. Note that the algorithm should never outputs ERROR.

Lemma 3. *Given input $r : S \mapsto \mathbb{F}_q$ and x, the Berlekamp-Welch algorithm returns f as defined in Definition 2.*

Proof. Let E, B be the respective polynomials found by step 1. We know $E(x)r(x) = B(x)$ for all $x \in S$. So for all $x \in S \backslash U$, we have $E(x)f(x) = B(x)$ as $r(x) = f(x)$ in this case. That is $E(x)f(x)$ and $B(x)$ have $|S \backslash U| = n - c$ points of agreement.

Suppose $Ef \neq B$. Since B has degree at most $n - c$, it must be the degree of Ef is at least $n - c$. This is a contradiction as $k + c < n - c$ given $c = \frac{n-k}{2} - 1$. Therefore, $Ef = B$ as required. □

2 Decoding Concatenation Codes

Given received word r of the following form

r_1	r_2	...	r_{n_1}

where each $r_{n_1} \in \Sigma_2^{n_2}$. If less than $\frac{d_1 d_2}{4}$ errors occur in r, then the number of blocks with at least $\frac{d_2}{2}$ errors is at most $\frac{d_1}{2}$.

The high-level algorithm for decoding concatenation codes is as follows:

1. Decode each r_i to the nearest $c_i \in C_2 \simeq \Sigma_1$.
2. Decode $(c_1, \ldots, c_{n_1})$ to the nearest codeword in C_1.

This decoding algorithm is efficient if there is an efficient decoding algorithm for C_1. Note that we can decode the inner blocks using brute-force in polynomial time w.r.t. the total length $n = n_1 n_2$.

Combine with the construction of asymptotically good codes using code concatenation technique from previous lecture, there are asymptotically good codes that can be efficiently encoded and decoded. This is summarized below.

Theorem 4. *There is an asymptotically good code over* $\{0,1\}$ *with* $\text{poly}(n)$ *time construction of generating matrix and* $\text{poly}(n)$ *time decoding from* $\Omega(1)$ *fraction of error.*

3 Elies-Bassilygo Bound on Rate vs δ

In previous lectures, we have seen several upper bounds on Rate of an Error R Correcting-code over binary alphabet in terms of relative distance δ. The Hadamard bound gave us $R \leq 1 - H(\delta/2)$, where $H(p) = p \log_2 \left(\frac{1}{p}\right) + (1-p) \log_2 \left(\frac{1}{(1-p)}\right)$, while Plotkin bound gave us $R \leq 1 - 2\delta$. While Hadamard is a better Upper bound for smaller δ, for larger δ, Plotkin bound is better. The best known lower bound is the GV-bound as $R \geq 1 - H(\delta)$. We will now look to prove an upper bound, which is better than both Hadamard and Plotkin bounds for all δ.

Hamming (Volume packing) bound is obtained using the fact that for an error correcting code, ball of radius $\frac{\delta}{2}$ are disjoint.

Theorem 5 (Elies-Bassilygo bound)**.** *In codes of distance* δ*, balls of radius* $\frac{\delta}{2} + \epsilon$ *are mostly disjoint. Using this, the following upper bound is obtained for binary codes,*

$$R \leq 1 - H(J(\delta)) + o(1)$$

where $J(\delta) = \frac{1 - \sqrt{1-2\delta}}{2}$ *is Johnson radius.*

To prove the upper bound, we will first look at the Johnson bound which shows that for a code C of distance δ, any ball of radius $\frac{\delta}{2} + \epsilon$ contains $\leq O(n)$ many codewords of C. This is proved using techniques similar to the Plotkin bound, which showed that a code with $\delta \geq 1/2$ can have $\leq O(n)$ codewords.

Theorem 6 (Johnson Bound)**.** *If $c_1, \ldots, c_L \in \{0,1\}^n$ such that $\Delta(c_i, c_j) \geq \delta n, \forall\ i \neq j$ and $r \in \{0,1\}^n$ such that $\Delta(r, c_i) \leq J(\delta)n, \forall\ i$ then $L \leq O(n)$.($\Delta(x, y)$ denotes the Hamming distance between x, y)*

Proof. We will first convert all c_i's from $\{0,1\}^n$ to v_i's $\{-1,+1\}^n$. The relation $\Delta(c_i, c_j) \geq \delta n, \forall\ i \neq j$ translates to $\langle v_i, v_j \rangle \leq (1-2\delta)n, \forall\ i \neq j$. Also, r will be converted in to $u \in \{-1.+1\}^n$ such that $\langle u, v_i \rangle \geq (1 - 2J(\delta))n, \forall\ i.(\langle x, y \rangle$ denotes the inner product of x and $y)$

Consider the set $S = \{v_i - \alpha u, i \in [L]\}$, for $\alpha = 1 - 2J(\delta)$. We look at the inner product.

$$\begin{aligned}
\langle v_i - \alpha u, v_j - \alpha u \rangle &= \langle v_i, v_j \rangle - \alpha(\langle v_i, u \rangle + \langle u, v_j \rangle) + \alpha^2 \langle u, u \rangle \\
&\leq (1 - 2\delta)n - 2\alpha(1 - 2J(\delta))n + \alpha^2 n \\
&= (\alpha^2 - 2\alpha^2 + \alpha^2)n \\
&= 0
\end{aligned}$$

Thus, the inner product of any 2 vectors in S is ≤ 0. From the claim in proof of Plotkin bound in lecture 2(Claim 3), we know that $|S| \leq 2n$. Thus, $L \leq O(n)$. □

Now we will prove the Elies-Bassilygo bound using the Johnson Bound.

Proof of Theorem 5. Take a code C of distance δ, rate R. From the Johnson bound, we have balls of radius $J(\delta)$ around the codewords cover each point of $\{0,1\}^n \leq O(n)$ times, i.e.

$$|C||B(J(\delta)n)| \leq 2^n \cdot O(n)$$

Substituting $C = 2^{Rn}$ and $|B(J(\delta)n)| \approx 2^{n(H(J(\delta))+o(1))}$, we have

$$2^{Rn} \leq \frac{2^n \cdot O(n)}{2^{n(H(J(\delta))+o(1))}}$$

Taking log both sides, we have

$$R \leq 1 - H(J(\delta)) + o(1)$$

□

4 Comparing the bounds on Rate vs δ

The following graph depicts how the various bounds compare on their estimate for Rate with increasing δ.

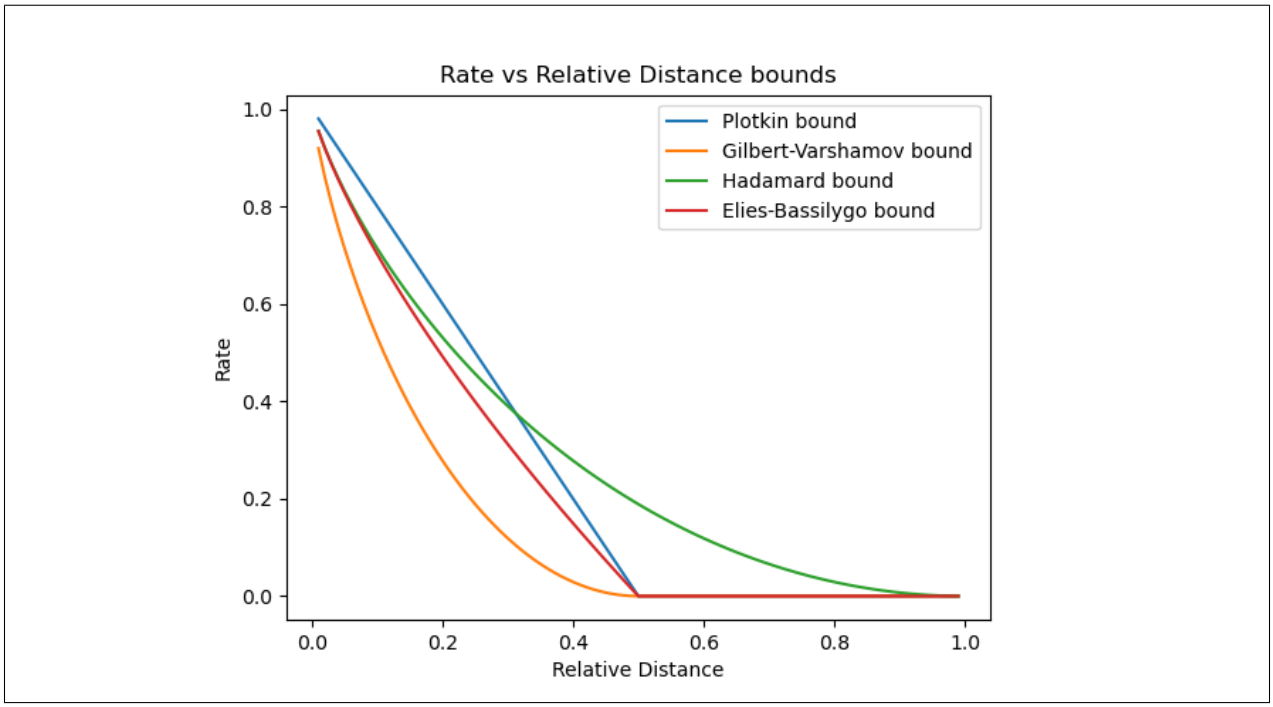

Figure 1: Various bounds for Rate vs Relative Distance

We will try approximating the bounds and draw a comparison between the different bounds.

Let $\delta = \frac{1-\gamma}{2}$, $\gamma \in [-1, 1]$.

1. **Plotkin bound** The bound of $R \leq 1 - 2\delta$ simply corresponds to $R \leq \gamma$.
2. **Gilbert-Varshamov bound** The lower bound of $R \geq 1 - H(\delta)$ becomes $1 - H\left(\frac{1-\gamma}{2}\right)$.

$$
\begin{aligned}
H\left(\frac{1-\gamma}{2}\right) &= \left(\frac{1-\gamma}{2}\right)\log_2\left(\left(\frac{2}{1-\gamma}\right)\right) + \frac{1+\gamma}{2}\log_2\left(\left(\frac{2}{1+\gamma}\right)\right) \\
&= 1 - \left(\frac{1-\gamma}{2}\right)\log_2(1-\gamma) - \left(\frac{1+\gamma}{2}\right)\log_2(1+\gamma) \\
&= 1 - \left(\frac{1-\gamma}{2}\right)\cdot c\cdot(-\gamma) - \left(\frac{1+\gamma}{2}\right)\cdot c\cdot\gamma + O(\gamma^3) \qquad \ln(1+t) = t - \frac{t^2}{2} + \frac{t^3}{3}\cdots \\
&= 1 - \frac{c\gamma^2}{2} + O(\gamma^3)
\end{aligned}
$$

where c is constant corresponding to conversion from $\log_2$ to ln, i.e. $\left(\frac{1}{\ln(2)}\right)$.

Substituting, we get that GV-bound approximates that there exists a code with rate $R \geq \frac{c\gamma^2}{2} + O(\gamma^3)$.

3. **Elies-Bassilygo bound** We have $R \leq 1 - H(J(\delta))$. We have $J(\delta) = \frac{1-\sqrt{1-2\delta}}{2} = \frac{1-\sqrt{\gamma}}{2}$.

   Using the approximation derived for GV-bound, we have $H(J(\delta)) = 1 - \frac{c\gamma}{2} + O(\gamma^{1.5})$. Therefore, for code C over $\{0,1\}^n$, rate $R \leq \frac{c\gamma}{2} + O(\gamma^{1.5})$.