Lecture: Decoding RS codes and An Introduction to RM Codes

In this lecture, we shall work towards a low-complexity algorithm for decoding RS codes. While any linear code can be decoded via standard array decoding (which is also MAP over the BSC), as seen earlier, such a decoding procedure has time (and space) complexity that is exponential in the blocklength n. Our objective is to come up with decoders that "work" in time $\approx$ poly(n).

Interpolating RS codes from erasures

Recall that the message vector $\underline{m} \in \mathbb{F}^K$ of an RS code is mapped to the polynomial

$$f(x) = m_0 + m_1 x + \cdots + m_{K-1} x^{K-1}$$

and then evaluated at points $\{\alpha_1, \cdots, \alpha_n\}$.

Now, suppose that upto $t \leq n-K = d-1$ erasures occur in a codeword $\underline{c} \in RS[n,K]$. Erasure decoding of RS codes is hence equivalent to interpolating a polynomial $(f(x))$ from its evaluations (at the unerased locations).

FACT: Any polynomial of degree $K-1$ is uniquely determined by its value at $n \geq K$ points.

HW: Prove the above fact using the fundamental theorem of algebra.

The reconstruction of $f(x)$ given $\geq K$ evaluations (since $\geq K$ positions are unerased) is carried out using the interpolation polynomials $(\rho_j(x) : j \in [K])$ as follows: Let $\{\alpha_1, \ldots, \alpha_K\}$ be the evaluation points corresponding to unerased locations. Then, define

$$\rho_j(x) = \prod_{\substack{i \neq j \\ i \in [K]}} \frac{x - \alpha_i}{\alpha_j - \alpha_i};$$

our interpolated polynomial is then $f(x) = \sum_{j=1}^{K} f(\alpha_j) \cdot \rho_j(x)$.

Decoding RS codes from errors

We now discuss an algorithm for decoding an RS codeword from $\leq \lfloor \frac{d-1}{2} \rfloor = \lfloor \frac{n-K}{2} \rfloor$ errors.

The following algorithm is due to [Welch-Berlekamp (1986)]. Let $y = (y_1, \ldots, y_n)$ be the noisy received vector obtained by corrupting $c = (c_1, \ldots, c_n)$ with $\leq \lfloor \frac{n-K}{2} \rfloor$ errors. Let $E = \{i : y_i \neq f(\alpha_i)\}$ be the set of error locations [unknown to the decoder!]. We define

$$E(x) = \prod_{i \in E} (x - \alpha_i)$$

to be the "error locator polynomial", of degree $\leq \lfloor \frac{n-K}{2} \rfloor$. Note that $E(x)$ has the property that $\forall i \in [n]$:

$$E(\alpha_i)\, y_i = E(\alpha_i) f(\alpha_i) \quad \text{[Why?]}$$

Further, define the polynomial $N(X) \triangleq E(X) f(X)$. Hence, the bivariate polynomial

$$P(X,Y) = E(X)Y - N(X)$$

satisfies $P(\alpha_i, y_i) = 0, \ \forall i \in [n]$. We use this observation to construct a decoding algorithm.

Algorithm (Welch-Berlekamp decoder):

① Identify a non-zero bivariate polynomial $Q(X,Y)$ that satisfies

$$Q(X,Y) = E_1(X)Y - N_1(X), \text{ for some } E_1(X), N_1(X) \text{ s.t.}$$

(i) $\deg(E_1(X)) \leq \left\lfloor \frac{n-k}{2} \right\rfloor$ and

(ii) $\deg(N_1(X)) \leq \left\lfloor \frac{n-k}{2} \right\rfloor + k - 1$,

(iii) $Q(\alpha_i, y_i) = 0, \ \forall i \in [n]$.

② Return $\hat{f}(X) = \frac{N_1(X)}{E_1(X)}$.

To prove that this decoder can correct $\leq \left\lfloor \frac{n-k}{2} \right\rfloor$ errors, we need to show the following lemma.

Lemma 1: (i) A non-zero solution Q to Step ① exists.

(ii) Any solution pair $(E(X), N_1(X))$ to Step ① must satisfy $f(X) = N_1(X)/E_1(X)$.

Proof: (i) Simply pick $E_1(x) = E(x)$ and $N_1(x) = N(x)$!

(ii) We define the polynomial $R(x) = E_1(x) f(x) - N_1(x)$. Note that

(a) $\deg(R(x)) \leq \left\lfloor \frac{n-k}{2} \right\rfloor + k - 1$, and

(b) $R(x)$ has at least $n - \left\lfloor \frac{n-k}{2} \right\rfloor$ roots: To see this, note that $\forall i \in [n]$ s.t. $y_i = f(\alpha_i)$, we have $R(\alpha_i) = 0$, by the definition of $Q(x,y)$.

Hence, if $n - \left\lfloor \frac{n-k}{2} \right\rfloor > \left\lfloor \frac{n-k}{2} \right\rfloor + k - 1$, we must have that $R(x) \equiv 0$ [why?], which indeed holds. Thus, $N_1(x)/E_1(x) = f(x)$. ▨

Remark: Step (i) can be carried out in poly(n) time, since one simply needs to solve a homogeneous system of linear equations, with the variables being the coefficients of $N_1(x), E_1(x)$.

Reed-Muller codes

We shall now discuss yet another (and possibly the most widely studied in recent years) family of linear codes based on polynomial evaluations: the Reed-Muller (RM) family of codes. Just as RS codes were obtained via evaluations of univariate polynomials of fixed maximum degree over points in a finite field, the RM family of codes is obtained via the

evaluations of multivariate polynomials of fixed maximum degree.

In particular, we shall be concerned with the family of binary RM codes, obtained as evaluations of polynomials $f(X_1, \cdots, X_m) \in \mathbb{F}_2[X_1, \cdots, X_m]$.

FACT: Any polynomial $f(X_1, \cdots, X_m) \in \mathbb{F}_2[X_1, \cdots, X_m]$ in m variables can be uniquely expressed as a sum (over $\mathbb{F}_2$) of monomials of the form $X_S \triangleq \prod_{i \in S} X_i$, for some $S \subseteq [m]$.

Remark: We do not need to consider monomials with variables raised to a power > 1, since $x^2 = x$, over $\mathbb{F}_2$.

Def 1: The degree of a monomial X_S is simply $|S|$.

Def 2: The degree of a polynomial $f(X_1, \cdots, X_m) = \sum_{S \subseteq [m]} a_S X_S$, where $a_S \in \{0,1\}$, $\forall S \subseteq [m]$, is simply $\max\{|S| : a_S = 1\}$, i.e., the maximum degree of a monomial in its expansion.

Def 3: For a fixed order $r \in [0:m]$, the r^{th}-order RM code is obtained as

$$RM(m,r) = \{ \underbrace{(f(z) : z \in \{0,1\}^m)}_{\triangleq \text{Eval}(f)} : \deg(f(X_1, \cdots, X_m)) \leq r \}.$$