CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 18: 13-10-2022

Scribe: Amit Rajaraman *Lecturer: Rohit Gurjar*

In this lecture, we elaborate a bit more on local decoding. Recall our discussion on Reed-Solomon codes. Are these locally decodable?
Given a polynomial

$$m = a_0 + a_1 x + \cdots + a_{d-1} x^{d-1}$$

and $(m(\alpha_1), m(\alpha_2), \ldots, m(\alpha_n))$, can we recover a specific a_i by looking at a few of the $m(\alpha_i)$? They do not seem very suitable for local decoding, so let us look at some other codes that are more amenable to this.

Definition 18.1 (Reed-Muller code)**.** *Let $\mathbb{F}$ be a finite field, and $\ell, d \in \mathbb{N}$ such that $|\mathbb{F}| > d$. Also fix $S_1, \ldots, S_\ell \subseteq \mathbb{F}$ The message space of the* Reed Solomon code $\mathrm{RM}(n, \ell, d)$ *is*

$$\{p(x_1, \ldots, x_\ell) \in \mathbb{F}[x_1, \ldots, x_\ell] : \deg(p) \leq d\}.$$

A polynomial p is encoded as

$$\mathrm{RM}(p) = (p(\alpha_1, \ldots, \alpha_\ell))_{\alpha \in (S_1 \times \cdots \times S_\ell)}.$$

In our setting, we fix all the S_i to be $\mathbb{F}$.
That is, the encoding goes from $\mathbb{F}^{\binom{d+\ell}{\ell}}$ to $\mathbb{F}^{|\mathbb{F}|^\ell}$. What is the distance of this code? Given a nonzero polynomial p over ℓ variables of degree at most d, what is the largest number of zeros it can have?

Proposition 18.2. *Any polynomial $p \in \mathbb{F}[x_1, \ldots, x_\ell]$ of degree at most d has at most $d|\mathbb{F}|^{\ell-1}$ zeros. That is,*

$$\Pr_{\alpha \sim \mathbb{F}^\ell}[p(\alpha) = 0] \leq \frac{d}{|\mathbb{F}|}$$

Proof. Assume wlog that the degree of p is d. The idea is that we will partition $\mathbb{F}^\ell$ into several "lines" and show that on each line, the probability is at most $d/|\mathbb{F}|$. For $\alpha \in \mathbb{F}^\ell, r \in \mathbb{F}^\ell$, consider the line

$$L_{\alpha,r} = \{\alpha + tr : t \in \mathbb{F}\}.$$

We shall show that for some clever choice of r, the polynomial does not become the zero polynomial on this line for any α. Restricted to this line, the function becomes a polynomial in t. We would like to show that this is a nonzero polynomial

$$p(\alpha_1 + tr_1, \ldots, \alpha_\ell + tr_\ell).$$

in t. Let P_d be the degree d part of P, and note that the coefficient of t^d in this polynomial is $P_d(r_1, \ldots, r_\ell)$, *independent of* α! Further, P_d cannot be identically zero on $\mathbb{F}^\ell$ because this would imply that the degree of p is less than d (this uses the fact that $|\mathbb{F}| > d$). Therefore, the polynomial is nonzero for some choice of r. This means that the univariate polynomial is nonzero, so has at most $d/|\mathbb{F}|$ zeros, and we are done. □

Are the Reed-Muller codes locally decodable? Let us change our perspective slightly, changing the message space from the coefficients to the evaluations of $\mathbb{F}$ at some $\binom{\ell+d}{d}$ (fixed and specific) points – there exists a choice of such points which uniquely determines the polynomial.
In the absence of errors, this makes local decoding trivial. What do we do in the presence of errors?
Suppose we want to evaluate the polynomial at some point β given the evaluations at all points (with an ϵ fraction of errors). If we manage to come up with some line through β that has relatively few errors, then we can use Reed-Solomon decoding on this line to compute what $p(\beta)$ is precisely. Suppose that we choose this line randomly. Then, the expected number of errors is

$$\begin{aligned}\mathbb{E}_{\text{random line } \ell \text{ through } \beta}[\text{number of corruptions on } \ell] &= \mathbb{1}_{\text{error at } \beta} + \frac{|\mathbb{F}| - 1}{|\mathbb{F}|^\ell - 1}(\text{number of errors not at } \beta) \\ &\leq 1 + \epsilon|\mathbb{F}|.\end{aligned}$$

Therefore, by a Markov argument,

$$\Pr_{\ell}[\ell \text{ has less than } 3(\epsilon|\mathbb{F}|+1) \text{ errors}] \geq \frac{2}{3}$$

and we are done.

In all, we choose a random line through β, apply Reed-Solomon coding on this line, then use the resultant polynomial to compute $p(\beta)$.
Here, the local decoding algorithm runs in $O(|\mathbb{F}|)$ time, which we wish to be $\mathsf{polylog}(|\mathbb{F}|^{\ell})$. For sufficiently large ℓ ($\Omega((|\mathbb{F}|/\log|\mathbb{F}|)^{\delta})$ for some constant $\delta > 0$), this is indeed true.

When we try to convert this to the binary setting however, one major issue pops up. We can of course view $\mathbb{F}$ as a string over $\{0,1\}^{\log|\mathbb{F}|}$, but in this case the notion of "error" changes. An ϵ fraction corruption means that an ϵ fraction of the *bits* are corrupted, not points in $\mathbb{F}^{\ell}$. Indeed, an ϵ fraction of bits being corrupted means that an $\epsilon \log|\mathbb{F}|$ fraction of the points in $\mathbb{F}^{\ell}$ could be corrupted.
We would like a coding scheme over the binary alphabet that can tolerate a constant fraction of errors, and Reed-Muller codes do not seem to satisfy this.