CS 229r Essential Coding Theory Apr 15, 2020

Lecture 21

Instructor: Madhu Sudan *Scribe: Eric Bornstein*

1 Overview

In this class we give constructions of locally correctable codes. We start with a code of rate 1/2. Then, we show how to get high rate codes that are locally correctable, which was first shown in [**?**]. Then, we use this to give locally correctable codes for any rate which are locally correctable almost up to half the distance. This distance reaches the Singleton Bound. This was first shown in [**?**].

2 Previous Best Results

First, we recall the definition of locally correctable codes.

Definition 1 (Locally Correctable/Decodable Codes)**.** A code $C \subset \{f : [n] \to \Sigma\}$ is a $(\ell, \varepsilon)-$LCC (locally correctable code) if there exists a decoder D such that for any $g : [n] \to \Sigma, f \in C$ with $\delta(f, g) \leq \varepsilon$, then for each x, $\mathbf{Pr}(D^g(x) \neq f(x)) \leq 1/3$ and D makes at most ℓ queries to g.

In this class we are interested in $\ell = o(n)$. Last class, we looked at constant ℓ. In fact, we will construct a $\ell = O(n^c)$ for any c.

Before the results of [**?**], the best known rates were from Bivariate Reed-Muller codes. Specifically, set $n = q^2, [n] \simeq \mathbb{F}_q^2, d = (1 - 2\varepsilon)q, C = \{f : \mathbb{F}_q^2 \to \mathbb{F}_q | \deg(f) \leq d\}$. Here, were are letting the degree of the polynomials be slightly less than q.

As we have seen, $\dim(C) = \binom{d+1}{2} \approx (1 - 2\varepsilon)^2 q^2/2 \approx (1 - \varepsilon')n/2$ for some ε'.

The local decoder D is similar to the one discussed last class. On input x and oracle g, the local decoder picks a random line $(x+tb)$ through x. Then, the decoder takes $g(x+tb)$ for all values of t. For any codeword f, $f(x + tb)$ is a polynomial of degree at most d in t, so if f and g agree on this line, the decoder can find the polynomial using $q - 1$ values on the line. This determines the value of $f(x)$. We have $\ell = q = \sqrt{n}$.

Notice that $R < 1/2$ and as R approaches 1/2, $\delta(C)$ approaches 0.

We give the full theorem here:

Theorem 2 (Locally Correctability of RM Codes)**.** *For any ε and $n = q^2$, there exists a Reed-Muller code with dimension $(1 - \varepsilon)n/2$ with distance at least ε. There also exists a local decoder that can decode up to half the distance using $O(\sqrt{n})$ elements.*

Exercise 3. *Show that if we use t variables, that the rate of the code will be at most $\frac{1}{t!}$ and that the value of ℓ will be about $n^{1/t}$.*

Proof. The rate claim is due to the rate of the t variable Reed-Muller code. For local correctability, we need q points on a line and $n = q^t$.

■

There is no clear way to generalize this to get a better rate than 1/2.

3 Multiplicity Codes

We will use multiplicity codes to get higher rates, which was done in [**?**]. We first show:

Theorem 4 (Local Correctability of Bivariate Multiplicity codes)**.** *Bivariate multiplicity codes of length n can locally correct a positive fraction of errors using $O(\sqrt{n})$ coordinates. The rate of these codes can go up to $2/3$.*

Our starting example will be the Bivariate Multiplicity 2 code. This is similar to a Reed-Muller. In addition to encoding the evaluation of a polynomial at several points, we also include the evaluation of the derivatives of the polynomial. Here, if $f(x,y) = \sum c_{i,j}x^iy^j$, then $f_x(x,y) = \sum ic_{i,j}x^{i-1}y^j$ and $f_y(x,y) = \sum jc_{i,j}x^iy^{j-1}$. The derivatives are defined in the same way as over $\mathbb{R}$. However, higher order derivatives are slightly different over $\mathbb{F}_{q^r}$ for $r > 1$.

Formally, the encoding of $f : \mathbb{F}_q^2 \to \mathbb{F}_q$ is $\langle f(a,b), f_x(a,b), f_y(a,b)\rangle_{a,b}$. This means that the alphabet $\Sigma = \mathbb{F}_q^3$ because we have three values. There are q^2 possible values for a and b. So, our code maps polynomials of degree at most d to $(\mathbb{F}_q^3)^{q^2}$. We still have $\dim_q(C) = \binom{d+1}{2} \approx d^2/2$, but now our alphabet is three times as large, so $\dim(C) \approx d^2/6$.

So far, the changes have not been useful. The benefit of our change is that we can now make $d = (1-2\varepsilon)2q$, which is twice as big as before. So, our rate is $\dim(C)/n \approx \frac{d^2/6}{n} \approx \frac{4q^2/6}{q^2} = 2/3$, which is better than before.

3.1 Decoding

Now we will show how to locally correct. This will prove that the code has a positive relative distance. The idea is similar to that of locally correctable Reed-Muller codes.

To find the value for input a and oracle g, we first pick a random line $a+tb$. We define $g_{a,b}(t) = g(a+tb)$. If f is a nearby codeword, $\deg(f) \leq d < 2q$. Note, we can compute $g'_{a,b}(a+tb)$ at any point by combining $g_x(a+tb)$ and $g_y(a+tb)$. This gives us q evaluations of the polynomial $g_{a,b}$ and the same number of its polynomial. This is enough to determine the polynomial because the polynomial has degree less than $2q$.

To see this, consider functions f and g that agree on q points and have derivatives that agree on those same points. This means that $f-g$ must have zeroes on those points. Because the derivatives agree, $f-g$ must have zeroes of multiplicity at least 2 on those points. This means that $f-g$ has at least $2q$ zeroes (double counting the zeroes). Because $\deg(f-g) < 2q$, we know that $f-g = 0$ and $f = g$.

This will allow us to find $g_{a,b}(a)$ and $g'_{a,b}(a)$. However, we need to find $g_x(a)$ and $g_y(a)$. We can do this by taking another random line $a+tc$ and finding $g'_{a,c}(a)$. Using a linear transformation, we can use these to calculate $g_x(a)$ and $g_y(a)$.

Note that we expect a constant fraction of the points on the line to be errors. So, we need to be able to find g even though there are some errors on the line. This leads us to the following exercise.

Exercise 5. *Show how to decode the univariate multiplicity 2 code from a positive fraction of errors. Use abstract decoding. The univariate multiplicity code has messages $f : \mathbb{F}_q \to \mathbb{F}_q$ and codewords $< f(a), f'(a) >_a$.*

Proof. We use abstract decoding similarly to the regular Reed-Solomon decoding. First note that our field elements are of the form $(a,a') \in \mathbb{F}_q$ where the first coordinate represents the evaluation of a polynomial and the second coordinate represents the evaluation of the derivative. Now, $(a,a') * (b,b') = (ab, ab' + a'b)$, where the second coordinate is using the product rule. We let $\mathcal{E}$ is the univariate multiplicity 2 code with polynomials of degree at most $2\varepsilon q$ and $\mathcal{W}$ is the same for degree at most $d + 2\varepsilon q$. We know that $\mathcal{E}$ has dimension $2\varepsilon q + 1$ and $\mathcal{W}$ has distance $2q - (d + 2\varepsilon q) = 2\varepsilon q$. Lastly, we see $\mathcal{E} * \mathcal{C} \subset \mathcal{W}$. We then apply abstract decoding.

■

3.2 Higher rate from Higher Multiplicity

We can extend this idea to codes with higher order derivatives. A multiplicity m code encodes the evaluation of the function and all its derivatives up to the $(m-1)$th derivatives. For decoding, this requires m lines to find the $m-1$st derivatives.

Exercise 6. *Show that the bivariate multiplicity m code has rate $1 - \frac{1}{1+m}$ and $\ell = O(m\sqrt{n})$*

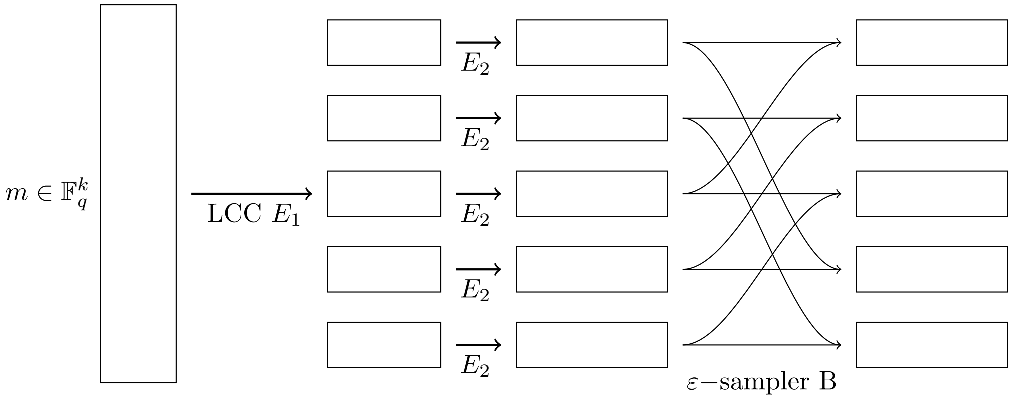

Figure 1: Construction of Locally Correctable Code

Proof. To calculate the rate, we need to know two numbers: the number of derivatives and the max degree d. Because $f_{xy} = f_{yx}$, the number of derivatives is the number of pairs of whole numbers (a, b) with $a + b < m$. This is $\binom{m+1}{2}$. The max degree is mq. The rate is $\frac{\binom{d+1}{2}}{\binom{m+1}{2}q^2} \approx \frac{m}{m+1} = 1 - \frac{1}{m-1}$.

To calculate locally decode, we use the same strategy of picking a random line. Now, to calculate the $(m-1)$th derivatives, we need m distinct lines.

■

3.3 Better Locality from More Variables

For any $\varepsilon > 0$, we can use $t = 1/\varepsilon$ variables (instead of two). For any constant m, this gives us $\ell = O(n^{1/t}) = O(n^{\varepsilon})$. If we let $m = 1/\varepsilon^2$, we also get rate approaching 1.

4 Applying AEL

Once we have good locally correctable codes for rates approaching 1 and a positive fraction of errors, we can use AEL to get good locally correctable codes for any rate. These codes will achieve the Singleton Bound. This was first shown in [**?**].

We start with a message m. Then, we apply E_1, which is a locally correctable code from the previous section with high rate, dimension k, and $\ell = k^{o(1)}$. For each coordinate of $E_1(m)$, we apply E_2, which is a code of rate R and distance $1 - R - o(1)$. Note, E_2 does not need to be locally correctable. Lastly, we use an $\varepsilon-$sampler B to mix the codewords. See Figure 1 for the construction diagram.

To correct any coordinate in the codeword, we need to know the corresponding value on the left side of the $\varepsilon-$sampler, which is in some E_2 codeword. To know this, we need to know what message was encoded. To know this, we can use the local correctibility of E_1. This requires knowing $k^{o(1)}$ of the E_2 messages. We can get these if we can decode the corresponding E_2 codewords (with errors). We can get these from the right side of the $\varepsilon-$sampler.

Now, we will trace the errors to see that this works. If there are $\frac{1-R-\varepsilon'}{2}$ errors, all but ε of the E_2 codewords will have about $\frac{1-R-\varepsilon'}{2}$. Because of this, we will be able to correctly decode all but the ε fraction of codewords. This was all we needed to locally correct any of the coordinates.