CS 229r Essential Coding Theory Apr 15, 2020

Lecture 21

Instructor: Madhu Sudan *Scribe: Hari Kothapalli*

1 Today's Topics

- Multiplicity codes: get good locality at any rate, we care mostly about high rates.
- Codes at the Singleton Bound: application of the AEL codes

2 Review

2.1 Locally Correctable Codes

Note that we will view codewords as functions mapping a coordinate to a letter of the alphabet, i.e. $C \subseteq \{[n] \mapsto \Sigma\}$.

Definition 1. *A code $C \subseteq \{[n] \mapsto \Sigma\}$ is (ℓ, ε)-locally correctable if $\exists$ decoder D such that for all*

1. *$g : [n] \mapsto \Sigma$ such that $\exists f \in C$ with $\delta(f, g) \leq \varepsilon$ (i.e. g close to some codeword in C)*
2. *$x \in [n]$*

$\Pr[D^g(x) \neq f(x)] \leq \frac{1}{3}$, where D^g makes ℓ queries to g.

N.B. we want ℓ to be sub-linear in n. We will achieve $\ell(n) = n^{o(1)}$.

2.2 Bivariate Reed-Muller Codes

The highest possible rate we have seen so far is bounded under $R < \frac{1}{2}$. To achieve rate arbitrarily close to $\frac{1}{2}$, we could use a family of bivariate codes.

Let $n = q^2$ ($[n] = \mathbb{F}_q \times \mathbb{F}_q \approx \mathbb{F}_q^2$) and $d = (1 - 2\varepsilon)q$. The Bivariate Reed-Muller code over $\mathbb{F}_q$ is

$$C = \{f : \mathbb{F}_q^2 \mapsto \mathbb{F}_q | \deg(f) \leq d\}.$$

From this, we see that

$$\dim(C) = \binom{d+1}{2} \approx \frac{(1-2\varepsilon)^2 q^2}{2} \approx \frac{(1-\varepsilon')}{2} n,$$

indicating that the rate of these codes approaches $\frac{1}{2}$.

Note on local-decoding: choosing $d < q$ is necessary for the decoding algorithm, as discussed in the previous lecture, to work. In this case, $\ell = q = \sqrt{n}$. Thus, we have sublinear locality, but only for rates $R < \frac{1}{2}$. In fact, for $\ell = n^{1/m}$, the upper bound on rate becomes $R < \frac{1}{m!}$. Multiplicity codes will allows us to achieve higher rates.

3 Multiplicity Codes [10.1145/2629416]

Theorem 2. *The bivariate multiplicity-m code achieves rate $R \approx 1 - \frac{1}{1+m}$ and is locally-correctable with $\ell = O_m(\sqrt{n})$.*

3.1 Encoding

Once again, let us consider the bivariate case; specifically the bivariate multiplicity-2 code. The general idea is that the encoding of a polynomial includes not only its evaluations but also **evaluations of its derivatives**. The encoding function E given f, a polynomial with degree less than d, outputs

$$E(f) = \langle (f(a,b), f_x(a,b), f_y(a,b)) \rangle_{a,b \in \mathbb{F}_q}$$

From this, we have $\Sigma = \mathbb{F}_q^3$ and $n = q^2$. $\dim(C) = \binom{d+1}{2} \approx \frac{d^2}{2}$ *field values*, which is really $\frac{d^2}{6}$ elements of Σ. This means that for $d = (1-2\varepsilon) \cdot 2q$, which is a factor of 2 improvement over what we have seen previously, we can still have high distance and local-decodability.

3.2 Local Decoding

First, let us consider why the local-decoding algorithm from last lecture does not quite work. Given some point $a \in F_q^2$, we want our decoder D^g to output $D^g(a) = f(a)$. Recall that D_g makes $\ell = q$ queries to g, such that on some line passing through the point a, we have the correct evaluation of g everywhere except for a. However, $\deg(g) \leq d < 2q$, which means that this is not enough information to uniquely identify the polynomial g.

In order to reconstruct g, we must use its derivative information. For every point on the line, we have the partial derivatives of g in the x and y direction – enough compute the derivative in the direction of the line. More formally, for some random point $b \in \mathbb{F}_q^2$, we define $g_{q,b}(t) \triangleq g(a+tb)$; we know the value $g_{a,b}(t)$ as well as $g'_{a,b}(t)$ for all $t \neq 0$. We claim that this is enough information to uniquely identify g.

Lemma 3. *If two polynomials f, g, such that $\deg(f) = \deg(g) \leq d$, and their derivatives f', g' agree on q points where $d < 2q$, then $f = g$.*

Proof. Consider the polynomial $p = f - g$; it is of degree at most d. If $f(x) = g(x)$ and $f'(x) = g'(x)$, then $p(x) = p'(x) = 0$. We say that x is a zero of multiplicity 2 for the polynomial p. (Aside: this is where the name "multiplicity codes" comes from). Since there are q points where both f and g agree and f' and g' agree, p has q zeros of multiplicity 2; thus p has at least $2q > d$ zeros, and is therefore identically zero. Finally, $f - g = 0 \implies f = g$. □

Note that we have been relying on the fact that for all sampled points, g is equal to our original polynomial f. Given that some constant ε fraction of the space is corrupted, on at least a few sample points g will likely differ from f. It turns out that we actually have enough information to correct some $\varepsilon' > 0$ fraction errors.

Exercise 4. *(Abstract Decoding) Show that even with some ε'-fraction corruptions, we can recover $g_{a,b}(0)$, given $g_{a,b}$ and $g'_{a,b}$. Hint: See the Welch-Berlekamp algorithm for Reed-Solomon decoding.*

However, we have not yet shown that we can recover the x and y partial derivatives of f, which are also part of its encoding. Recall that we have $g'_{a,b}$, i.e. the derivative of g in the direction of some line passing through a and a random point b. Let us pick another random point $b' \in \mathbb{F}_q^2$ and construct $g_{a,b'}$ and therefore $g'_{a,b'}$. By a union bound, if $g'_{a,b}(0)$ and $g'_{a,b'}(0)$ are individually correct with high probability, they are both correct with almost as high probability (note that for higher multiplicity codes, we are still dealing with a constant number of lines but the union bound is over more variables). Now, using a basis shift, we can compute $g_x(a)$ and $g_y(x)$ from $g'_{a,b}(0)$ and $g'_{a,c}(0)$. Thus, the full local-decoding algorithm requires $\ell = 2q$ queries.

3.3 Rate Analysis

We have $k \approx \frac{d^2}{6}$, $n = q^2$, and, treating ε as a negligible constant, $d \approx 2q \cdot (1-2\varepsilon)$:

$$k \approx \frac{2}{3} q^2 \implies R \to \frac{2}{3}.$$

Thus, we have beaten the previous bound of $R = \frac{1}{2}$! By increasing the multiplicity of the code, we can achieve even higher rates. For example, consider the bivariate multiplicity-3 code. In this case,

$$E(f) = \langle (f(a,b), f_x(a,b), f_y(a,b), f_{xx}(a,b), f_{xy}(a,b), f_{yy}(a,b)) \rangle_{a,b \in \mathbb{F}_q}$$

And so, $\Sigma = \mathbb{F}_q^6$, $k \approx \frac{d^2}{12}$, $n = q^2$, and $d \approx 3q$, from which we get

$$k \approx \frac{9}{12} q^2 \implies R \to \frac{3}{4}.$$

In general for bivariate multiplicity-m codes, we have $\Sigma = \mathbb{F}_q^{\binom{m+1}{2}}$, $k \approx \frac{d^2}{2 \cdot \binom{m+1}{2}}$, $n = q^2$, $d \approx mq$, implying that $R \approx 1 - \frac{1}{m+1}$. Thus, as $m \to \infty$, $R \to 1$.

3.4 Improving Locality

The locality of these codes as described above is $\ell = O_m(\sqrt{n})$. For $\varepsilon > 0$, we can achieve n^ε locality using $t = \frac{1}{\varepsilon}$ variables. This works for any constant multiplicity m. Thus, taking $m = \frac{1}{\varepsilon^2}$ or something even larger, we get rate R tending towards 1. Note that the tradeoff here is not between locality and rate but rather between locality and fraction of errors we can correct.

4 Codes approaching the Singleton Bound [10.1145/3051093]

From here, we will treat the multiplicity codes as a black-box in our next construction. Namely, we require only that there exists codes with rate $R \to 1$ and locality n^ε.

Now, for arbitrary rate R, we want to achieve

- distance: $1 - R - \varepsilon$
- locality: $n^{o(1)}$
- fraction errors we can correct: $\frac{1-R-\varepsilon'}{2}$

4.1 Encoding

To this end, we will apply the AEL construction. Remember that this construction first encodes messages using some high rate encoding E_1. In this case, we will use the multiplicity codes which can correct $o(1)$-fraction errors with locality $k^{o(1)}$. We then apply another encoding E_2 on each symbol of the result of E_1. We pick E_2 has rate R and distance $1 - R - o(1)$, which are guaranteed to exist over sufficiently large alphabets. Finally, we permute the bits according to some ε-sampler bipartite graph B, which encoding we will call E_B.

4.2 Decoding

Assume $\frac{1-R-\varepsilon'}{2}$ fraction of symbols were corrupted. By the properties of an ε-sampler, when we invert the permutation E_B, only a small fraction of the blocks will have greater than $\frac{1-R-\varepsilon'}{2}$-fraction errors. We can correctly decode all other blocks using the decoder for E_2, since they will have fraction errors less than half the distance of E_2. Now, when we go to decode E_1, only a small fraction of symbols will be incorrect. Again, this decoding will work, since we know that E_1 can correct some $o(1)$-fraction of errors.

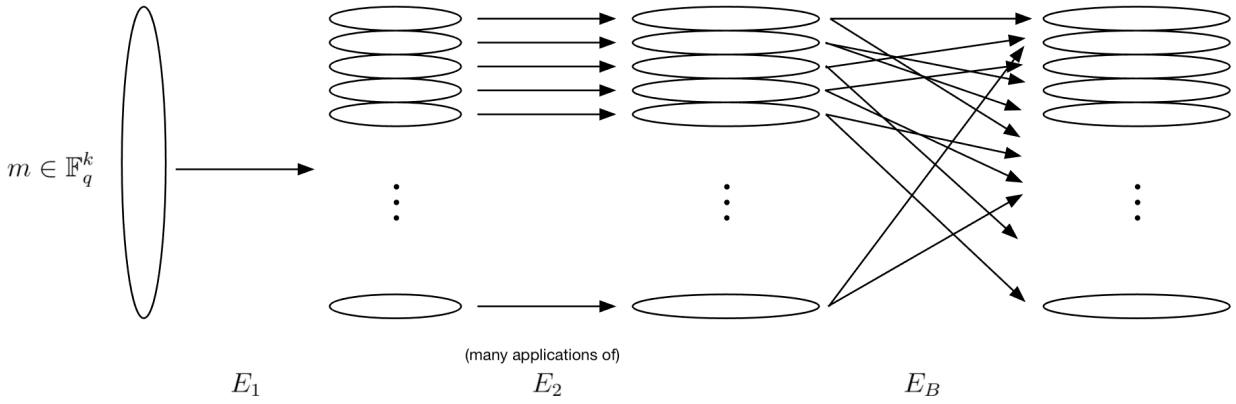

Figure 1: The AEL code construction. Note that E_1 is locally-correctable, while E_2 need not be.

4.3 Locality

Suppose some coordinate in the final codeword is an error; we want to determine its actual value. Inverting E_B, we can figure out which block its belongs to. Now, since E_2 is not locally-decodable, we need to determine the correct value of the entire block; note that this affects locality only by some negligible amount equal to the degree of B. In turn this means that we must locally-decode some output symbol of E_1, which we know that we can do in $o(1)$ (with some negligible factors given that we must backtrack through E_B and E_2 to determine the value of some output symbol of E_1).

Theorem 5. *There exist codes with locality $n^{o(1)}$ and rate R that can correct $\frac{1-R-\varepsilon'}{2}$-fraction errors, over an alphabet that grows with ε.*