Lecture 6: Fourier Analysis and The Rate/Distance Trade-off for Distance Near 1/2

Topics in Error-Correcting Codes (Fall 2022)
University of Toronto
Swastik Kopparty
Scribes: Brinda Venkataramani and Harry Sha

A quick recap of some facts:

Fact 1. *$R \geq 1 - H(\delta)$ (GV bound), where H is the BCE function.*

Fact 2. *$R \leq 1 - H(J(\delta))$ (EB bound), where J is the Johnson function $J(\delta) = \frac{1-\sqrt{1-2\delta}}{2}$.*

Fact 3. *There exist explicit codes with rate vs. δ being what comes out of concatenation of RS outer codes with brute-force search inner-codes. (This relies on GV bound since this is the best known lower-bound.)*

We will now focus on the case where $\delta = \frac{1-\epsilon}{2}$.

Note: In our discussion, we are first letting $n \to \infty$ and then only $\delta \to \frac{1-\epsilon}{2}$.

We know there exist codes with rate $\Omega(\epsilon^2)$. For all codes, rate is at most $O(\epsilon)$. This is quite a big gap. In this lecture we will see that the first bound is nearer the truth. This was first proved by McEliece, Rodemich, Rumsey and Welch in the 70s through their "Linear Programming bound". We will see a simple proof (that only works for linear codes) of this result for the δ close to 1/2 asymptotic. This proof is a translation to Fourier analysis of an eigenvalue-based proof by Alon and an improvement by Schectman and Shraibman.

Worth noting: Explicit codes in this regime. We also know there exist explicit codes with rate at least $\Omega(\epsilon^3)$. This comes from concatenation. The main question is how to choose distances d_1, d_2, s.t. $\delta = \frac{1-\epsilon}{2}$. The idea is that the inner brute-force code being binary forces it to have distances around $\frac{1}{2}$, i.e. not too far below. So we consider $\delta_{in} = \frac{1-O(\epsilon)}{2}$ and thus we can take $R_{in} = O(\epsilon^2)$ and for the outer code $\delta_{out} = 1 - O(\epsilon)$ so $R = \Omega(\epsilon^3)$.

Currently, in the field, it is believed that this gap between codes which can exist and those which are explicitly constructible should be arbitrarily small.

1 Fourier analysis on $\mathbb{F}_2$

We consider the space of all functions

$$\{f : \mathbb{F}_2^k \to \mathbb{R}\}.$$

This is a 2^k-dimensional vector space. A natural basis is:

$$\delta_x(y) = \begin{cases} 1 & y = x \\ 0 & \text{otherwise} \end{cases}.$$

i.e. 2^k indicator functions for each $x \in \mathbb{F}_2^k$.

However, this is not the only basis. We can introduce the Fourier basis:

Consider the functions $\psi_y(x) = -1^{\langle x,y\rangle}$, where the inner product is the $\mathbb{F}_2$ inner product, and $y \in \mathbb{F}_2^k$. Note this is indeed a well-defined map into $\mathbb{R}$ since the i.p. is modulo 2.

Lemma 4. *The space of ψ_y is an orthogonal space.*

Proof.

$$\begin{aligned} \langle \psi_y, \psi_{y'} \rangle &= \sum_{x \in \mathbb{F}_2^k} \psi_y(x)\psi_{y'}(x) \\ &= \sum_x -1^{\langle x,y\rangle} - 1^{\langle x,y'\rangle} \\ &= \sum_x -1^{\langle x,y+y'\rangle} \\ &= \sum_x \psi_{y+y'}(x) \\ &= \sum_x -1^{\sum\limits_{1\le i\le k} x_i z_i} && (z_i = y_i + y_i') \\ &= \prod (1 + (-1)^{z_i}) \end{aligned}$$

This is just the indicator function for $z_i = 0$ for some i. i.e.

$$\sum_x \psi_{y+y'}(x) = \begin{cases} 2^k & y + y' = 0 \\ 0 & \text{otherwise} \end{cases}$$

□

Fact 5. *For every nice finite abelian group there is such a nice Fourier basis respecting the group operation.*

This Fourier basis lets us write $f : \mathbb{F}_2^k \to \mathbb{R}$ as

$$f = \sum_y \hat{f}(y)\psi_y,$$

with

$$\hat{f}(y) = \langle f, \psi_y \rangle,$$

with scaling $\frac{1}{2^k}$.

$\hat{f}$ is itself a function from $\mathbb{F}_2^k \to \mathbb{R}$ called the Fourier Transform (FT) of f.

Fact 6 (Parseval's identity)**.** *$\langle f, g\rangle = \langle \hat{f}, \hat{g}\rangle$ by orthogonality of ψ_y for different y with the appropriate scaling.*

This FT will be useful for studying codes in $\mathbb{F}_n^2$. Notice we could have defined the FT directly on $\mathbb{F}_2^n$ but it will be cleaner this way since the original message space is $\mathbb{F}_2^k$.

2 Bounding the rate

Let C be a linear code in $\mathbb{F}_2^n$. The goal is to express the distance in terms of FTs. Let

$$G = \begin{pmatrix} \vdots & \vdots & \cdots & \vdots \\ v_1 & v_2 & \ddots & v_n \\ \vdots & \vdots & \cdots & \vdots \end{pmatrix},$$

be a generator matrix for C with all v_is distinct. We then consider

$$f_i = \begin{cases} \frac{1}{n} & x = v_i \\ 0 & \text{otherwise} \end{cases}$$

We will start by studying $\hat{f}_i$. We have:

$$\begin{aligned} \hat{f}_i(y) &= \sum_x f_i(x)\psi_y(x) \\ &= \frac{1}{n}\sum \psi_y(v_i) \\ &= \frac{1}{n}\sum (-1)^{\langle v_i, y\rangle} \end{aligned}$$

We can view this result as follows:

Consider the set of all x so $\langle x, y\rangle = 0$. This is a subspace of $\mathbb{F}_2^k$. Also, we can consider the orthogonal complement of x so the i.p. is 1. In this view the final term above is a distribution of how the v_i are distributed among these subspaces.

Recall C is a code with distance δ. Consider y_G a codeword. Then to pull out the coordinates of y_G w.r.t. G we take the i.p. $\langle y_G, v_i\rangle$. Notice that y_G has at least δ 1's since weight measures distance of a code. Also, $\delta = \frac{1-\epsilon}{2}$. So at most $1 - 2\delta \leq \epsilon$ 1's. We will use this fact to show that n is big relative to k.

i.e. The goal is to upper bound the rate in terms of distance where the distance $\delta = \frac{1-\epsilon}{2}$ is close to $1/2$. We want $R \leq \epsilon^2$, equivalently, $n \geq k/\epsilon^2$.

Definition 7. *A code has small bias if distance is bounded above and below.*

For now, we will let C be such a code. We will also show that it is possible to remove this condition. Under this, using similar reasoning, we can get that $\hat{f}_i(y) \geq -\epsilon$, so $\hat{f}_i(y) \leq |\epsilon|$. And also $\hat{f}(0) = 1$. The trick is to use Parseval's identity.

The support of f is at least L_1-norm. i.e.

$$\text{support} f \geq \frac{\left(\sum |f(x)|\right)^2}{\sum |f(x)|^2} = \frac{1}{\sum |\hat{f}_i(y)|^2}.$$

(With the appropriate scaling.)

Lemma 8 (Linearity of ψ_y)**.**

$$\psi_y(a+b) = \psi_y(a)\psi_y(b)$$

So, we found

$$n = \text{support}(f) \geq \frac{2^k}{1 + \epsilon^2(2^k - 1)} \approx \frac{1}{\epsilon^2}.$$

This is not exactly what we wanted - we're missing a factor of k.

3 Improvement Using Convolution

To close the gap we will introduce a tool that allows us to shrink the Fourier coefficents.

Definition 9 (Convolution)**.** *Define the convolution of two functions f, $g : \mathbb{F}_2^k \to \mathbb{R}$, as*

$$f * g(z) = \sum_{x, x' \in \mathbb{F}_2^k : x + x' = z} f(x)g(x') = \sum_{x \in \mathbb{F}_2^k} f(x)g(z - x)$$

The main point is the following lemma.

Lemma 10. $f \hat{*} g(y) = \hat{f}(y)\hat{g}(y)$

Proof. Computing the Fourier coefficients of the convolution, we get

$$\begin{aligned} f \hat{*} g(y) &= \sum_{z \in \mathbb{F}_2^k} (f * g)(z)\psi_y(z) \\ &= \sum_{z \in \mathbb{F}_2^k} \sum_{x \in \mathbb{F}_2^k} f(x)g(z - x)\psi_y(z) \\ &= \sum_{z \in \mathbb{F}_2^k} \sum_{x \in \mathbb{F}_2^k} f(x)g(z - x)\psi_y(x)\psi_y(z - x) && \text{(by lemma 8)} \\ &= \left(\sum_x f(x)\psi_y(x)\right)\left(\sum_u g(u)\psi_y(u)\right) \\ &= \hat{f}(y)\hat{g}(y) \end{aligned}$$

□

Take the function $f = \frac{1}{n} \cdot \text{Indicator}\{v_1, ..., v_n\}$. Consider $h = f^{*t} = f * f ... * f$ (t times). Since $\hat{f}(0) = 1$, and $|\hat{f}(x)| \leq \epsilon$. By , $\hat{h}(0) = 1$, and $|\hat{h}(z)| \leq \epsilon^t$.

Lemma 11. *If f, g are probability mass functions (pmf) corresponding to independent random variables A and B, then $f * g$ is the pmf for $A+B$. In particular, $\sum f * g(x) = 1$, and* $\text{support}(f \cdot g) = \{a + b : a \in \text{support}(f), b \in \text{support}(g)\}$

Proof. This is essentially by the definition of convolution. For $f * g(z)$, the expression is the sum over all the possible ways for $A + B = z$, the probability that that happens. The claim about the support is true since it represents all the possible values attainable by $A + B$. □

By the same argument as before, we get

$$\begin{aligned}
\text{support}(h) &\geq \frac{\left(\sum |h(x)|\right)^2}{\left(\sum h(x)^2\right)} \\
&= \frac{1}{\frac{1}{2^k}\sum \hat{h}(y)^2} && \text{(by lemma 11)} \\
&\geq \frac{2^k}{1 + \epsilon^{2t}(2^k - 1)} \approx \frac{1}{\epsilon^{2t}}
\end{aligned}$$

Then $n^t = \text{support}(f)^t \geq \text{support}(h)$, so again, we have $n \geq 1/\epsilon^2$. To improve this, we will use a better bound on $\text{support}(f)$ in terms of $\text{support}(h)$.

Observation 12. $\text{support}(f^{*t}) \subseteq \{\sum_{x \in S} x : S \subseteq \text{support}(f)\}$.

Note that this follows immediately from lemma 11. So instead of bounding $\text{support}(f)^t \geq \text{support}(h)$, we will bound

$$\text{support}(h) \leq \binom{n}{0} + \binom{n}{1} + ... + \binom{n}{t} \leq \left(\frac{en}{t}\right)^t \cdot t \leq \left(\frac{cn}{t}\right)^t$$

for some constant c just a tiny bit bigger than e. Thus, we have

$$\begin{aligned}
t\left(\frac{en}{t}\right)^t &\geq \frac{2^k}{1 + \epsilon^{2t}(2^k - 1)} \\
\implies n &\geq \frac{1}{c}\left(\frac{2^k}{1 + \epsilon^{2t}(2^k - 1)}\right)^{1/t} t
\end{aligned}$$

Set t s.t. $\epsilon^{2t} = 1/2^k$, (i.e. $t = \frac{k}{2\log(1/\epsilon)}$). Plugging this back in, we get

$$
\begin{aligned}
n &\geq \frac{1}{c}\left(\frac{2^k}{2}\right)^{1/t} t \\
&\geq \Omega(2^{k/t} t) \\
&= \Omega\left(\frac{k}{\epsilon^2 \log(1/\epsilon)}\right),
\end{aligned}
$$

which is what we wanted (up to the additional $\log(1/\epsilon)$)!

4 Removing the bias assumption

We had $\hat{f}(y) \leq \epsilon$ for all y. I.e. $\hat{f}(y) \in [-1, \epsilon]$. In the previous part, we introduced the additional assumption that $\hat{f}(y) \geq -\epsilon$. We will address this here.

We have $f^{*3} \geq 0$, (since f was a probability measure).

$$
0 \leq f^{*3}(0) = \frac{1}{2^k} \sum_y (\hat{f^{*3}})(y) \psi_y(0) = \frac{1}{2^k} \sum_y (\hat{f^{*3}})(y)
$$

Where we used the fact that $\psi_y(0) = 1$ for all y. Thus, we have $\sum_y \hat{f^{*3}}(y) \geq 0$. Additionally, by lemma 10, we have $\sum_y \hat{f}(y)^3 \geq 0$. We will use this fact to show that

$$
A = \sum_{y:\hat{f}(y) < -\epsilon} \hat{f}(y)^2
$$

is small. We have

$$
\begin{aligned}
0 &\leq \sum_y \hat{f}(y)^3 \\
&= \sum_{y:\hat{f}(y) < -\epsilon} \hat{f}(y)^3 + 1 + \sum_{y:|\hat{f}(y)| \leq \epsilon} \hat{f}(y)^3 \\
&\leq (-\epsilon) \sum_y \hat{f}(y)^2 + 1 + \sum_{y:|\hat{f}(y)| \leq \epsilon} \hat{f}(y)^3 \\
&\leq -\epsilon A + 1 + \epsilon^3 (2^k - 1)
\end{aligned}
$$

So $A \leq \frac{1+\epsilon^3(2^k-1)}{\epsilon} = \frac{1}{\epsilon} + \epsilon^2(2^k - 1)$.

Applying this to h, we get

$$
\begin{aligned}
\sum_y \hat{h}(y)^2 &= 1 + A + \sum_{y:|\hat{h}(y)| \leq \epsilon^t} \hat{h}(y)^2 \\
&= 1 + \frac{1}{\epsilon^t} + 2\epsilon^{2t}(2^k - 1)
\end{aligned}
$$

Plugging this back into the analysis with a different choice in t allows us to problem the same bound within a constant factor.