CS 579: Computational Complexity Fall 2025

Lecture 03 - September 02

Prof. Fernando Granha Jeronimo *Super Scribes: Andrei Staicu & Pranav Rajpal*

1 Previous Lecture

When we discussed space-bounded computation through the problem of STConn with $G = (V, E)$ a directed graph, it captures non-deterministic computation. We saw Savitch's theorem:

Theorem 1.1 (Savitch's Theorem)**.**

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}((f(n))^2)$$

We can also study undirected connectivity, i.e., USTConn, which captures reversible computation, and it is a complete problem for SL (symmetric log-space).

We then did **Boolean Fourier Analysis**, the analysis of functions $f : \{0,1\}^n \to \mathbb{R}$. Depending on the setting it is useful to view the domain as $\mathbb{Z}_2^n$ or $\mathbb{F}_2^n$.

The characters of $\mathbb{Z}_2^n$ are as follows: $\forall S \subseteq [n]$, $\chi_S : \{0,1\}^n \to \mathbb{R}$ by $\chi_S(x) = \prod_{i \in S}(-1)^{x_i}$.

These functions form an orthonormal basis with inner product:

$$\langle f, g \rangle = \mathbb{E}_{x \in \{0,1\}^n}[g(x)f(x)] = \frac{1}{2^n} \sum_{x \in \mathbb{Z}_2^n} g(x)f(x) \tag{1}$$

We can write $f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x)$, and for any $f : \mathbb{Z}_2^n \to \mathbb{R}$ we can compute Fourier coefficients as follows:

$$\widehat{f}(S) = \mathbb{E}_x[f(x)\chi_S(x)] \tag{2}$$

1.1 Representation Theory Aside

A *representation* of a group G on a complex-vector space V is a homomorphism $\rho_V : G \to \text{GL}(V)$, if V is k-dimensional, then $\text{GL}(V)$ is the set of invertible $k \times k$ matrices with entries in $\mathbb{C}$.

A *sub-representation* of a representation V is a subspace $W \subset V$ invariant under G, $\rho_V(g)(W) \subset W$ for all $g \in G$. A representation is *irreducible* if it has no non-trivial sub-representations.

A map $\phi : V \to W$ (a matrix) is G-linear if for all $g, h \in G$, $\phi \cdot \rho_V(g) = \rho_W(g) \cdot \phi$. Note that everything in the previous expression is a matrix, so this is saying something analogous to: matrices commute. If ϕ is G-linear, then $\ker \phi$ and $\text{Im}\, \phi$ are both subrepresentations of V and W respectively.

Schur' Lemma says the following: if V, W are irreducible representations and $\phi : V \to W$ is G-linear either $\phi = \lambda I$ for some $\lambda \in \mathbb{C}$ or $\phi = 0$.

Proof. For any $\phi : V \to W$ that is G-linear, $\ker \phi$ is a sub-representation so must be zero. Since $\mathbb{C}$ is algebraically closed, ϕ has an eigenvalue so $\phi - \lambda I$ has a kernel, which also must be zero. Therefore $\phi = \lambda I$. □

In any finite abelian group G, (in particular $\mathbb{Z}_2^n$), all irreducible representations are 1-dimensional, and thus all isomorphic to $\mathbb{C}$. Since $\rho_V : G \to \mathbb{C}^*$ is a group homomorphism:

$$\rho_V(g) \cdot \rho_V(h) = \rho_V(gh) = \rho_V(hg) = \rho_V(h) \cdot \rho_V(g) \tag{3}$$

Each $\rho_V(g) \in \mathbb{C} \setminus \{0\}$ is G-linear, we have $\rho_V(g) = \lambda_{g,V}$.

The **character** of the irreducible representation (ρ_V, V), recall $\rho_V : G \to \mathrm{GL}(V)$, is the function $\chi_V : G \to \mathbb{C}$ by $g \mapsto \lambda_{g,V}$.

Notice also $\chi_V(g \cdot h) = \lambda_{gh,V} = \lambda_{g,V} \cdot \lambda_{h,V}$, since all elements of $\mathbb{Z}_2^n$ "square" to 1, then $\lambda_{g,V} = \pm 1$ for all $g \in \mathbb{Z}_2^n$ and all irreducible representations V have real eigenvalues.

The following facts were hinted at in class:

- If G is an abelian group, the characters of G^n are $\chi(g_1, \ldots, g_n) = \prod \chi_{V_i}(g_i)$ where V_i is any irreducible representation of G.
- In an abelian group, the number of irreducible representations is precisely the size of the group $|G|$.
- In an abelian group, characters of irreducible representations form an orthonormal basis over the vector space of all functions $f : G \to \mathbb{C}$, with respect to the inner product:

$$\langle \alpha, \beta \rangle = \frac{1}{|G|} \sum_{g \in G} \overline{\alpha(g)} \cdot \beta(g) \tag{4}$$

In $\mathbb{Z}_2^n$, the **characters** are explicitly $\chi_S(x) = \prod_{i \in S} (-1)^{x_i}$ for $x \in \mathbb{Z}_2^n$, this statement follows from the first fact, and that the only representations of $\mathbb{Z}_2$ are $\rho_V(1) = 1$ or $\rho_V(1) = -1$.

Check out *Representation Theory* by Fulton and Harris [4], Chapters 1 and 2 for the proofs of the above statements, however these are way beyond the scope of this class.

---

2 Coding Theory

Coding theory deals with the following problem: we want to start with a message x, $|x| = k$, encode into c, $|c| = n$, where if $\tilde{c}$ is a corrupted message, x can be recovered. We want to do the procedure while: handling errors, recovering information about x, and minimizing the overhead introduced.

Definition 2.1. *A **code** C of block length n over an alphabet Σ is $C \subseteq \Sigma^n$. Elements $c \in C$ are **codewords**.*

Definition 2.2. *The **Hamming distance** between $x, y \in \Sigma^n$ is the number of places x and y disagree:*

$$\Delta(x, y) := \sum_{i=1}^{n} \mathbf{1}_{x_i \neq y_i} \tag{5}$$

*The **relative Hamming distance** is $\delta(x, y) := \frac{1}{n}\Delta(x, y)$ and the **minimum distance** of a code $C \subseteq \Sigma^n$ is $\min_{c \neq c'} \Delta(c, c')$.*

Codes with larger minimum distance can correct more errors.

Definition 2.3. *The **message length** of a code C over an alphabet Σ is defined to be*

$$k = \log_{|\Sigma|} |C| \tag{6}$$

or equivalently, the message length k is the number of symbols from Σ needed to represent any arbitrary message.

Definition 2.4. *The **rate** of a code C is the ratio of its message length to its block length, i.e., k/n. Informally, this represents how much longer a message becomes when encoded using a specific code, so a larger rate means less overhead is introduced.*

Consider all linear functions over $\mathbb{F}_2[x_1, \ldots, x_n]$ with zero constant. This is simply:

$$f(x) = \sum_{i=1}^{n} c_i x_i \tag{7}$$

where the addition happens $\mod 2$ and $c_i \in \mathbb{F}_2 = \{0, 1\}$. Thus, every linear function is of the form

$$f_S(x) = \bigoplus_{i \in S} x_i, \quad \text{for any subset } S \subseteq [n] \tag{8}$$

Consider the map that takes $g \in \mathbb{F}_2[x_1, \ldots, x_n]$ linear, to the vector of its evaluations over all of $\mathbb{F}_2^n$, denote this by $\text{Eval}(f)$. This encoding is an injective map that takes $\mathbb{F}_2^n \to \mathbb{F}_2^N$ where $N = 2^n$.

Define the **Hadamard code** by the set of encodings of linear functions, i.e. $\{\text{Eval}(l_S)\}_{S \subset [n]}$. First we can define a linear code:

Definition 2.5. *A **linear code** C of dimension k and block length n over a finite field $\mathbb{F}$ is a k-dimensional subspace of $\mathbb{F}^n$.*

Proposition 2.6. *If C is a linear code, then the minimum distance of C is $\min_{x \in C} \Delta(x, 0)$.*

Proof. Say the minimum distance of C, which we'll call d, is the minimum distance between some distinct $x, y \in C$, meaning that $d = \Delta(x, y)$. Since C is a linear code, $x - y \in C$ as well, and the number of positions that x disagrees with y will be the same as the number of positions that $x - y$ disagrees with 0 because $(x-y)_i \neq 0$ if and only if $x_i \neq y_i$. That means that $d = \Delta(x-y, 0)$, which tells us that the minimum distance can be represented as $\Delta(c, 0)$ for some $c \in C$, as desired. □

The Hadamard code has the following properties:

- It is a linear code
- It has block length 2^n
- It has message length n
- It has minimum distance 2^{n-1}.

This implies that it has rate $n/2^n$ and relative distance $1/2$. The only statement that doesn't follow from the definition is its distance.

Proposition 2.7. *The Hadamard code has minimum relative distance $1/2$.*

Proof. As shown in Proposition 2.6, we can find the minimum relative distance of a linear code C by finding $\min_{x \in C} \delta(x, 0)$. In the case of the Hadamard code, 0 refers to the all zero vector, which is the evaluation vector of $f_\emptyset$, which means that $\Delta(x, 0) = \Delta(f_S, f_\emptyset)$ is the number of inputs x such that $f_S(x) = 1$.

The idea that we can use here is that if we have 2 codewords corresponding to the functions $f_S, f_T : \mathbb{F}_2^n \to \mathbb{F}_2$ then the value of $(-1)^{f_S(x)} \cdot (-1)^{f_T(x)} = (-1)^{f_S(x)+f_T(x)}$ is dependent only on whether $f_S(x)$ and $f_T(x)$ are equal:

- If $f_S(x) = f_T(x) = c$ for some $c \in \{0, 1\}$ then $c + c = 2c \equiv 0 \pmod 2$, which means that $(-1)^0 = 1$.
- If $f_S(x) \neq f_T(x)$ then you'd get that $f_S(x) + f_T(x) = 1$ so $(-1)^1 = -1$.

That lets us easily relate the distance between the codewords f_S and $f_\emptyset$ as a sum that ends up being easier to work with:

$$\sum_{x \in \mathbb{F}_2^n} (-1)^{f_S(x)} \cdot (-1)^{f_\emptyset(x)} = \sum_{f_S(x)=1} (-1) + \sum_{f_S(x)=0} 1$$

In order to find the distance $\Delta(f_S, f_\emptyset)$, we can evaluate the sum on the left by noticing that the character function $\chi_S(x)$ for any $S \subseteq [n]$ can be written in terms of $f_S(x)$ as

$$\chi_S(x) = (-1)^{f_S(x)} \tag{9}$$

which can be seen fairly easily by expanding out $f_S(x)$ as a sum mod 2.

That means that the sum above can be written as an inner product:

$$\sum_{x \in \mathbb{F}_2^n} (-1)^{f_S(x)} \cdot (-1)^{f_\emptyset(x)} = \sum_{x \in \mathbb{F}_2^n} \chi_S(x)\chi_\emptyset(x) = 2^n \langle \chi_S, \chi_\emptyset \rangle \tag{10}$$

Since $\langle \chi_S, \chi_\emptyset \rangle = 0$ (due to the character functions forming an orthonormal basis), then

$$\sum_{f_S(x)=1} (-1) + \sum_{f_S(x)=0} 1 = 0 \tag{11}$$

That can only happen if exactly half of the evaluation points of f_S have $f_S(x) = 1$, which is equivalent to saying that $\delta(f_S, f_\emptyset) = \frac{1}{2}$, as desired. □

2.1 Locally Testable Codes

One natural property testing question that can be asked about a code is whether it is possible to of test if any arbitrary string is close to a codeword. That's extremely useful, since if we received a message that was encoded using a given code, we'd want to be able to tell whether it had been corrupted with some errors that we'd need to try correcting, or if it was still a valid codeword. For our purposes, we're going to focus on fast probabilistic tests of how likely it is that we'll be able to correct all the errors in a provided word. The motivation here is that if the string that we received ends up having too many errors for us to be able to correct them, then we'll need to ask the sender to send the message again (in the hopes that it won't get corrupted as much the second time), but the full algorithm for trying to decode the received string into a message may be time-consuming, so we'd like to be able to quickly test whether waiting for decoding to finish is worthwhile [3].

Codes where the above goal is feasible are called locally testable codes, which, informally speaking, are codes such that, using only a few queries to a given word $x \in \Sigma^n$, you can check if x is close to a codeword $c \in C$ (i.e. x is either the same as c or a corrupted version of c with relatively few errors). More formally, we can define locally testable codes as follows.

Definition 2.8. *A code $C \subseteq \Sigma^n$ is **locally testable** [1, 3] with query complexity $q \in \mathbb{N}$ and soundness $\eta > 0$ if there exists some randomized testing algorithm that takes in as input some $x \in \Sigma^n$ and either accepts or rejects. That algorithm needs to meet the following properties:*

- *If $x \in C$, then $\Pr[\text{test accepts}] = 1$ (i.e. the tester always accepts for valid codewords).*
- *If $x \notin C$, then $\Pr[\text{test rejects}] \geq \eta \cdot \delta(x, C)$ where $\delta(x, C)$ is the minimum relative distance between x and any codeword in C. Importantly, the test becomes more likely to reject as the relative distance (and consequently the number of errors) increases.*
- *The testing algorithm can only query at most q bits of the input string x.*

To help illustrate that definition of locally testable codes, we can show that the Hadamard codes discussed above are locally testable codes. In that case, the word that we'd receive as input to our algorithm would be a vector in $\mathbb{F}_2^N$ which represents the evaluation $\text{Eval}(f)$ for some $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Being able to test if f is a valid codeword means that we'd want to check if the provided function f is linear, since Hadamard codes are by definition the set of all linear functions, while at the same time minimizing the number of entries in the input vector we'd need to read. It's worth noting that since the vector that we want to test is the evaluation vector of a function f, reading out the bit at some arbitrary position x in $\text{Eval}(f)$ is equivalent to calling f with the value x. Because of that, we interchangeably treat the input string for our testing algorithm as a vector in $\mathbb{F}_2^N$ and as an oracle $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that we can make queries to.

In order to show that Hadamard codes are locally testable, we will first need to define what a convolution is and some relevant properties, since that will be useful in our analysis.

Definition 2.9. *The convolution of f, g as functions over $\mathbb{F}_2$ is defined by*

$$(f * g)(x) = \mathbb{E}_{y \in \mathbb{F}_2^n} f(y)g(x+y) = \frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} f(y)g(x+y)$$

Proposition 2.10. *If f, g are two functions $\mathbb{F}_2^n \to \mathbb{C}$, then $\widehat{(f * g)}(S) = \widehat{f}(S) \cdot \widehat{g}(S)$.*

Proof. This is mostly just algebraic manipulation using Equation 2 and the definition of convolution:

$$\begin{aligned}
\widehat{(f*g)}(S) &= \langle (f*g), \chi_S \rangle \\
&= \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (f*g)(x)\chi_S(x) \\
&= \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} \left(\frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} f(y)g(x+y) \right) \chi_S(x) \\
&= \frac{1}{2^{2n}} \sum_{x,y \in \mathbb{F}_2^n} f(y)g(x+y)\chi_S(x) \\
&= \frac{1}{2^{2n}} \sum_{u,y \in \mathbb{F}_2^n} f(y)g(u)\chi_S(u+y) && u = x+y \implies x = u+y \\
&= \frac{1}{2^{2n}} \sum_{u,y \in \mathbb{F}_2^n} f(y)g(u)\chi_S(u)\chi_S(y) && \chi_S \text{ is a homomorphism} \\
&= \left(\frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} g(u)\chi_S(u) \right) \left(\frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} f(y)\chi_S(y) \right) \\
&= \langle g, \chi_S \rangle \cdot \langle f, \chi_S \rangle \\
&= \hat{g}(S) \cdot \hat{f}(S)
\end{aligned}$$

□

Using that, we can then show that Hadamard codes are locally testable.

Proposition 2.11. *Hadamard codes are locally testable codes with query complexity $q = 3$ and soundness $\eta = 1$.*

Proof. Consider the following test [2]: sample $x, y \in \mathbb{F}_2^n$ uniformly and accept if and only if $f(x+y) = f(x) + f(y)$. We know that if $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is linear, then

$$f(x+y) = f(x) + f(y), \quad \forall x, y \in \mathbb{F}_2^n$$

It's fairly easy to see that that test accepts for all linear functions f, but if f is nonlinear, the test has some probability of accepting if it happens to pick values of x and y such that $f(x+y) = f(x) + f(y)$ even if the entire function isn't linear. Intuitively, we'd expect that a function with fewer errors would be more likely to accept, since we'd be more likely to pick x and y such that we read locations that haven't been corrupted, but we want to formalize that idea to show that $\Pr[\text{test rejects}] \geq \delta(x, C)$.

First define $g(x) = (-1)^{f(x)}$. Note that if $f(x+y) = f(x) + f(y)$, then $(-1)^{f(x+y)+f(x)+f(y)} = 1$ else it is -1, thus to translate this into the probability the test accepts, we do the following.

$$\Pr[\text{test accepts}] = \frac{1}{2^{2n}} \sum_{x,y \in \mathbb{Z}_2^n} \mathbb{1}_{f(x)+f(y)=f(x+y)} = \mathop{\mathbb{E}}_{x,y \in \mathbb{Z}_1^n} \left[\frac{1}{2} + \frac{1}{2}(-1)^{f(x+y)+f(x)+f(y)} \right] \tag{12}$$

$$= \frac{1}{2} + \frac{1}{2} \mathop{\mathbb{E}}_{x \in \mathbb{Z}_2^n} \left[\mathop{\mathbb{E}}_{y \in \mathbb{Z}_2^n} g(x+y)g(x)g(y) \right] = \frac{1}{2} + \frac{1}{2} \mathop{\mathbb{E}}_{x \in \mathbb{Z}_2^n} [g(x) \cdot (g*g)(x)] \tag{13}$$

Notice that by the definition of the convolution and the inner product, we reduce this to the sum over all the Fourier coefficients.

$$\Pr[\text{test accepts}] = \frac{1}{2} + \frac{1}{2} \sum_{S \subset [n]} \widehat{g}(S)\widehat{g}(S)\widehat{g}(S) \leq \frac{1}{2} + \frac{1}{2} \max_S |\widehat{g}(S)| \langle g, g \rangle = \frac{1}{2} + \frac{1}{2} \max_S |\widehat{g}(S)| \tag{14}$$

where we're using above that $\langle h, g\rangle = \mathbb{E}[f(x)g(x)]$ and, since g only takes $\{-1, 1\}$ then $\mathbb{E}_x g(x)^2 = 1$, and therefore $\langle g, g\rangle = 1$. Notice also that $\widehat{g}(S) = \langle g, \chi_S\rangle$. Suppose g and χ_S disagree at $\epsilon 2^n$-many places (or equivalently that $\delta(g, \chi_S) = \epsilon$), then we have that:

$$\langle g, \chi_S\rangle = \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} g(x)\chi_S(x) = \frac{1}{2^n}\left((2^n - \epsilon 2^n) - \epsilon 2^n\right) = 1 - 2\epsilon \tag{15}$$

Thus we get that $2\Pr[\text{test accepts}] \leq 1 + 1 - 2\epsilon$ so $\epsilon \leq 1 - \Pr[\text{test accepts}]$, or equivalently $\Pr[\text{test rejects}] \geq \epsilon$. We know that $\chi_S(x) = (-1)^{f_S(x)}$ and that $g(x) = (-1)^{f(x)}$, which means that $\chi_S(x) = g(x)$ if and only if $f_S(x) = f(x)$, implying that $\delta(f, f_S) = \delta(g, \chi_S)$, and therefore that $\Pr[\text{test rejects}] \geq \delta(f, f_S)$, as desired. □

References

[1] "Locally testable code (LTC)". In: *The Error Correction Zoo*. Ed. by Victor V. Albert and Philippe Faist. 2022. URL: https://errorcorrectionzoo.org/c/ltc.

[2] M. Blum, M. Luby, and R. Rubinfeld. "Self-testing/correcting with applications to numerical problems". In: *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*. STOC '90. Baltimore, Maryland, USA: Association for Computing Machinery, 1990, pp. 73–83. ISBN: 0897913612. DOI: 10.1145/100216.100225. URL: https://doi.org/10.1145/100216.100225.

[3] Irit Dinur et al. *Locally Testable Codes with constant rate, distance, and locality*. 2021. arXiv: 2111.04808 [cs.IT]. URL: https://arxiv.org/abs/2111.04808.

[4] William Fulton and Joe Harris. "Representations of Finite Groups". In: *Representation Theory: A First Course*. New York, NY: Springer New York, 2004, pp. 3–25. ISBN: 978-1-4612-0979-9. DOI: 10.1007/978-1-4612-0979-9. URL: https://doi.org/10.1007/978-1-4612-0979-9.