1 Last time

1.1 Overview

Last time we covered learning parity functions and $k$-juntas. We also went through the basics of Fourier analysis for functions $f: \{-1, 1\}^n \to \mathbb{R}$.

Today we finish the basics of Fourier analysis over the Boolean cube and cover two learning algorithms for Boolean functions, the LMN (Linial/Mansour/Nisan) algorithm and the KM (Kushilevitz/Mansour) algorithm. As we will see, the LMN algorithm lets us learn to high accuracy given uniform random labeled examples if we are given a set $S$ of subsets of $[n]$ whose Fourier coefficients have almost all of the “Fourier weight” of $f$. The KM algorithm uses membership queries and can be used to find all the “heavy” Fourier coefficients of $f$.

1.2 Basics of Fourier analysis, concluded

We recall that $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S(x)$ where $f: \{-1, 1\}^n \to \mathbb{R}$. We define $\chi_S(x) = \prod_{i \in S} x_i$ (note that $\chi_S(x)$ is the multilinear monomial corresponding to the product of all variables in $S$). We also defined $\hat{f}(S) = E[f(x)\chi_S(x)] = \langle f, \chi_S \rangle$. For Boolean $f$, this is equal to $\Pr[f(x) = \chi_S(x)] - \Pr[f \neq \chi_S]$.

As an example of the Fourier representation of a Boolean function, let us consider the AND function.

$$\text{AND}(x_1, \ldots, x_k) \overset{\text{def}}{=} \begin{cases} -1 & \text{if } x_1 = \cdots = x_k = -1 \\ +1 & \text{otherwise.} \end{cases}$$

To determine the Fourier representation of AND, first let’s consider the real-valued function $f$ such that $f(x_1, \ldots, x_k) = 1$ if $x_1 = \cdots = x_k = -1$, and 0 otherwise. It’s not
hard to see that

\[ f(x) = \frac{1-x_1}{2} \cdot \frac{1-x_2}{2} \ldots \frac{1-x_k}{2} = \sum_{S \subseteq [k]} (-1)^{|S|} \chi_S(x) \]

(note that \( \frac{1-x_j}{2} \) is either 0 or 1 when \( x_j \in \{-1, 1\} \)). Then we see that

\[ \text{AND}(x) = 1 - 2f(x) = 1 - \frac{2}{2^k} \sum_{|S| > 0} (-1)^{|S|} \chi_S(x). \]

Turning back to a general Boolean function \( f : \{-1, 1\}^n \to \{-1, 1\} \), it is easy to see from the definition that \( |\hat{f}(S)| \leq 1 \). However, a much stronger statement is in fact true: for any \( f : \{-1, 1\}^n \to \{-1, 1\} \), we have that \( \sum_{S \subseteq [n]} \hat{f}(S)^2 = 1 \). This is a special case of the following more general result:

**Theorem 1** (Plancherel’s Identity). For any two functions \( f, g : \{-1, 1\}^n \to \mathbb{R} \), we have

\[ \sum_{S \subseteq [n]} \hat{f}(S)\hat{g}(S) = \langle f, g \rangle = \mathbb{E}[fg]. \tag{1} \]

**Proof.** By plugging in the Fourier expansion of \( f \) and \( g \) in the RHS, using linearity of expectation:

\[
\mathbb{E}[fg] = \mathbb{E}_{x \sim \mathcal{U}\{-1,1\}^n} \left[ \left( \sum_S \hat{f}(S)\chi_S(x) \right) \left( \sum_T \hat{g}(T)\chi_T(x) \right) \right] = \sum_{S,T} \hat{f}(S)\hat{g}(T) \cdot \mathbb{E}_x [\chi_S(x)\chi_T(x)] = \sum_S \hat{f}(S)\hat{g}(S)
\]

where the last equality derives from the orthonormality of the \( \chi_S \)'s.

**Corollary 2** (Parseval’s Theorem). Let \( f : \{-1, 1\}^n \to \mathbb{R} \) be a Boolean function. Then

\[ \sum_{S \subseteq [n]} \hat{f}(S)^2 = \|f\|_2^2 \]

and in particular, if \( f \) is Boolean then \( \sum_{S \subseteq [n]} \hat{f}(S)^2 = 1 \).
1.3 Fourier and learning

The intuition is that the $\hat{f}(S)^2$’s are the “weights” of the $\chi_S$’s in the Fourier representation of $f$; so that learning a big enough fraction of the total weight means learning a “good fraction” of the function (in a sense that we shall make precise). Hence, our goal in Fourier-based learning is to approximate most of the Fourier representation of $f$.

A basic first objective is, given $f$, $S \subseteq [n]$, to find $\hat{f}(S)$. We note that $\hat{f}(S) = E[f(x)\chi_S]$. Clearly, computing this quantity exactly requires $2^n$ queries; however, we do not need the exact value! Hence, we will be satisfied with an approximation of $\hat{f}(S)$, which we can obtain efficiently via sampling.

Lemma 3. Fix any Boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$. Given $S \subseteq [n]$, $\gamma > 0, \delta > 0$ and access to independent, random examples $(x, f(x))$ where $x \sim U\{-1,1\}^n$, one can, using $O(1/\gamma^2 \log(1/\delta))$ such examples, output a value $c_S$ such that with probability greater than or equal to $1 - \delta$, $c_S$ satisfies $|c_S - \hat{f}(S)| \leq \gamma$.

Proof. Follows from the Hoeffding bound (additive Chernoff) on the random variable $f(x)\chi_S(x)$ (where $x$ is uniform random over $\{-1,1\}^n$), whose expected value is $\hat{f}(S)$.

Note that this random variable always takes values in $[-1,1]$ and hence we may straightforwardly apply the Hoeffding bound.

Remark 1. This is useful, but does not solve the learning question: there are $2^n$ Fourier coefficients, and for a Boolean function almost all of them are very small in magnitude. For instance, at most $n^4$ subsets $S \subseteq [n]$ are such that $\sum_{S \in S} \hat{f}(S)^2 \geq 1/n^2$. This is because otherwise we would get $\sum_S \hat{f}(S)^2 > n^4 \cdot 1/n^4 = 1$, contradicting Parseval’s theorem.

But suppose we were given a list of subsets $S_1, \ldots, S_m$ with the promise that 99% of the “Fourier weight” is on these: $\sum_{i=1}^m \hat{f}(S_i)^2 \geq 0.99$. Then we would be in good shape, as we could use the lemma above to estimate each $\hat{f}(S_i)$ to sufficiently high accuracy. The next section will make this precise.

2 The LMN Algorithm (Linial–Mansour–Nisan)

2.1 Preliminary Definitions

Definition 4. Fix $f : \{-1,1\}^n \rightarrow \{-1,1\}$, $S = \{S_1, \ldots, S_m\}$ where $S_i \subseteq [n]$. We say that $f$ is $\epsilon$-concentrated on $S$ if

$$\sum_{S_i \in S} \hat{f}(S_i)^2 \geq 1 - \epsilon$$
(that is, \( \sum_{S \notin S} \hat{f}(S)^2 \leq \epsilon \)).

### 2.2 LMN Algorithm

**Algorithm 1 The LMN Algorithm**

**Require:**
- access to random samples \((x, f(x))\)
- collection of subsets \(S = \{S_1, \ldots, S_m\}\)
- parameters \(\tau > 0, \epsilon > 0\)

1: Pick \(M = O\left(\frac{m}{\tau} \log \frac{m}{\delta}\right)\) random examples \((x, f(x))\).

2: Use this set to get an estimate \(c_{S_i}\) of \(\hat{f}(S_i)\), for all \(i \in [m]\).

3: \textbf{return} \(h: \{-1, 1\}^n \rightarrow \mathbb{R}\) defined by \(h(x) = \sum_{i=1}^m c_{S_i} \chi_{S_i}(x)\).

**Theorem 5.** Suppose \(f\) is \(\epsilon\)-concentrated on \(S\). Then with probability \(\geq 1 - \delta\), the hypothesis \(h\) returned by the LMN Algorithm satisfies \(\mathbb{E}_{x}[(f(x) - h(x))^2] \leq \epsilon + \tau\).

**Proof.** Fix some \(S_i \in S\). Define \(\gamma = \sqrt{\tau/m}\). From the earlier lemma, we are guaranteed that the probability of \(|c_{S_i} - \hat{f}(S_i)| > \gamma|\) is at most \(\delta/m\). With probability \(\geq 1 - \delta\), by the union bound, we get \(|c_{S_i} - \hat{f}(S_i)| \leq \gamma|\) for all \(i \in \{1, \ldots, m\}\).

Let \(g \stackrel{\text{def}}{=} f - h\). We have

\[
\mathbb{E}[(f - h)^2] = \mathbb{E}[g^2] = \sum_{S \in \mathcal{S}} \hat{g}(S)^2 = \sum_{S \in \mathcal{S}} \hat{g}(S)^2 + \sum_{S \notin \mathcal{S}} \hat{g}(S)^2 \
\leq \epsilon + \tau
\]

where the latest inequality holds because \(|g(S_i)| = |\hat{f}(S_i) - \hat{h}(S_i)| \leq \gamma|\), which implies that \(\sum_{S_i \in \mathcal{S}} \hat{g}(S_i)^2 \leq m\gamma^2 \leq \tau\); and \(\sum_{S \notin \mathcal{S}} \hat{g}(S)^2 = \sum_{S \notin \mathcal{S}} \hat{f}(S)^2 \leq \epsilon\) since \(h\) has no nonzero Fourier coefficients outside \(\mathcal{S}\). \(\square\)

The following observation allows us to convert easily the output of the LMN algorithm (which is a real-valued function) to a Boolean function with the same approximation guarantees:
Observation 6. Let us fix any Boolean function \( f : \{-1, 1\}^n \rightarrow \{-1, 1\} \), and a real \( h : \{-1, 1\}^n \rightarrow \mathbb{R} \). Define
\[
h'(x) = \text{sign}(h(x)) = \begin{cases} 
1 & h(x) \geq 0 \\
-1 & h(x) < 0
\end{cases}.
\]
Then we have that \( \Pr[h'(x) \neq f(x)] \leq \mathbb{E}[(f(x) - h(x))^2] \).

Proof. Note that the left hand side evaluates to \( \frac{1}{2^n} \sum_x \mathbf{1}_{\{h'(x) \neq f(x)\}} \) (the indicator function) while the right hand side evaluates to \( \frac{1}{2^n} \sum_x (f(x) - h(x))^2 \). We will show the desired inequality holds term by term, which will prove the result: for any \( x \),
- if \( h'(x) = f(x) \), the corresponding summand on the LHS evaluates to 0 and while the RHS summand is \( \geq 0 \);
- if \( h'(x) \neq f(x) \), the LHS summand contributes 1 to the sum while the RHS summand contributes \( \geq 1 \).

\[\square\]

2.3 Summing Up

Theorem 7 (LMN algorithm). Given a collection of subsets \( S \) such that \( f \) is \( \epsilon \)-concentrated on \( S \), as well as parameters \( \epsilon, \delta > 0 \) and access to independent, uniform random examples \( (x, f(x)) \), the LMN algorithm runs in \( \text{poly}(n, |S|, 1/\epsilon, \log(1/\delta)) \) time and outputs a hypothesis \( h' : \{-1, 1\}^n \rightarrow \{-1, 1\} \) such that \( \text{dist}(h', f) \leq 2\epsilon \) with probability at least \( 1 - \delta \).

This is tremendously helpful, but requires that \( S \) be given. We discuss below some ideas on how to get \( S \):

1. For some \( \mathcal{C} \) (classes of functions), we can show that every \( f \in \mathcal{C} \) is \( \epsilon \)-Fourier concentrated on \( S = \{ S \subseteq [n] : |S| \leq d \} \). We see that \( |S| \simeq n^d \). Hence, we get \( \text{poly}(n^d, 1/\epsilon) \) time algorithms. We explore this approach in future lectures.

2. Using membership queries, we can find \( S \) such that \( |\hat{f}(S)| \) is large! Thus, we can learn concept classes \( \mathcal{C} \) which have \( 1 - \epsilon \) of their Fourier weight on such “big” coefficients. The Kushilevitz-Mansour (KM) algorithm, the subject of the next section, is an algorithm that lets us find \( S \) such that \( |\hat{f}(S)| \) is large using MQ.
3 The KM Algorithm (Kushilevitz–Mansour)

3.1 Introduction

In the KM algorithm, we find $S$ such that $|\hat{f}(S)|$ is large, using membership queries.

Want: given $\theta$, efficiently find $S \overset{\text{def}}{=} \{ S \subseteq [n] : |\hat{f}(S)| \geq \theta \}$. Sadly, this is unrealistic: since we can only ever hope to obtain approximate values of the Fourier coefficients, there is no way to handle the sets at or very close to the threshold.

Will get: given $\theta$ and MQ($f$), find $S$ s.t.

- if $|\hat{f}(S)| \geq 2\theta$, $S_i \in S$;
- if $|\hat{f}(S)| \leq \theta/2$, then $S_i \not\in S$.

with probability at least $1 - \delta$, in poly($n, 1/\theta, \log 1/\delta$) time.

High-level idea: We want to “successively isolate” large $\hat{f}(S)$’s. So we think of a way to break up the set of all $S \subseteq [n]$ as follows:

Definition 8. Fix $k \in \{0, \ldots, n\}$, and $S_1 \subseteq [k]$. Given $f : \{-1, 1\}^n \rightarrow \mathbb{R}$, we define $f_{k,S_1} : \{-1, 1\}^{n-k} \rightarrow [-1, 1]$ by

$$f_{k,S_1}(x) \overset{\text{def}}{=} \sum_{T_2 \subseteq \{k+1, \ldots, n\}} \hat{f}(S_1 \cup T_2)\chi_{T_2}(x).$$

(note that we will use the subscripts to indicate where the sets “live”: $S_1, T_1 \subseteq \{1, \ldots, k\}$, while $T_2 \subseteq \{k + 1, \ldots, n\}$).

Observe that $f_{k,S_1}$ “includes” (in a sense we will define shortly) exactly the Fourier coefficients $\hat{f}(S)$ such that $S \cap \{1, \ldots, k\} = S_1$. So all $f_{k,S_1}$ as $S_1$ ranges over all subsets of $\{1, \ldots, k\}$ includes all $2^n$ Fourier coefficients of $f$.

Example 9.\footnote{Note that from the definition it is not immediately obvious that the range of $f_{k,S_1}$ is $[-1, 1]$; however we will prove this below.}
• For $k = 0$, we have $f_{0,\emptyset}(x) = f(x) = \sum_{T_2 \subseteq \{1, \ldots, n\}} \hat{f}(T_2) \chi_{S_2}(x)$

• For $k = n$, $S_1 \subseteq n$, we have that $f_{n,S_1}(x) = \hat{f}(S_1)$ (constant function).

The question that begs asking: supposing we are given $MQ(f, k, S_1 \subseteq \{1, \ldots, k\}, x)$, can we compute or approximate $f_{k,S_1}(x)$?

$\leadsto$ Using the definition, yes (term by term) – but this would be insanely inefficient

$\leadsto$ there is a better way, by sampling – with the following lemma:

### 3.2 Building up to the KM algorithm

We propose a better way than the one given above.

**Lemma 10.** Fix $k \in \{0, \ldots, n\}$, $S_1 \subseteq [k]$, $x \in \{-1, 1\}^{n-k}$, $f$. Then

$$f_{k,S_1}(x) = E_{y \sim \{-1,1\}^k} [f(y) \chi_{S_1}(y)].$$

(where we wrote $yx$ for the concatenation $(y_1, \ldots, y_k, x_1, \ldots, x_{n-k})$)

**Proof.** We note that $f(yx) = \sum_T \hat{f}(T) \chi_T(yx)$. Writing $T$ as $T_1 \cup T_2$ (where $T_1 = T \cap \{1, \ldots, k\}, T_2 = T \cap \{k+1, \ldots, n\}$), we get $\chi_T(yx) = \chi_{T_1}(y) \chi_{T_2}(x)$. Hence, we see that

$$f(yx) = \sum_{T_1} \sum_{T_2} \hat{f}(T_1 \cup T_2) \chi_{T_1}(y) \chi_{T_2}(x).$$

Therefore, by linearity

$$E_{y \sim \{-1,1\}^k} [f(yx) \chi_{S_1}(y)] = \sum_{T_1} \sum_{T_2} \hat{f}(T_1 \cup T_2) \chi_{T_2}(x) E[\chi_{T_1}(y) \chi_{S_1}(y)]$$

$$= \sum_{T_1} \sum_{T_2} \hat{f}(T_1 \cup T_2) \chi_{T_2}(x) 1_{\{T_1 = S_1\}} \quad \text{(by orthonormality)}$$

$$= \sum_{T_2} \hat{f}(S_1 \cup T_2) \chi_{T_2}(x)$$

$$= f_{k,S_1}(x)$$

\[\square\]

Using this lemma, $f(yx) \chi_{S_1}(y)$ (with $y \sim \{-1,1\}^k$) is a $[-1,1]$-valued random variable we can sample (as long as $f$ takes values in $[-1,1]$), and invoking Hoeffding–Chernoff bounds we get:
Lemma 11. There is an algorithm which, given MQ($f$) (where $f : \{-1, 1\}^n \to \{-1, 1\}$), $k \in \{0, \ldots, n\}$, $S_1 \subseteq [k]$ and $x \in \{-1, 1\}^{n-k}$, as well as parameters $\gamma, \delta > 0$, outputs with probability at least $1 - \delta$ a value $v \in [-1, 1]$ satisfying

$$|v - f_{k,S_1}(x)| \leq \gamma$$

in time $\text{poly}(n, \frac{1}{\gamma}, \log \frac{1}{\delta})$, making $O\left(\frac{1}{\gamma^2} \log \frac{1}{\delta}\right)$ queries.

Simplification: for the sake of clarity, we hereafter pretend we can get the exact value $f_{k,S_1}(x)$, instead of just approximate it. This will make the exposition cleaner, and can be addressed modulo some technical details.

Recall that our ultimate goal is to find large $\hat{f}(S)$ if there is one. We first make a couple useful observations:

Observation 12. Given $k, S_1 \subseteq [k]$ as above, we have

$$\mathbb{E}[f_{k,S_1}(x)^2]_{(\text{Plancherel})} = \sum_{T_2 \subseteq \{k+1, \ldots, n\}} \hat{f}(S \cup T_2)^2 = \sum_{S \subseteq [k]} \hat{f}(S)^2.$$

Observation 13. Fix any $f : \{-1, 1\}^n \to \{-1, 1\}$ and $\theta > 0$. Then:

(i) At most $1/\theta^2$ sets $S \subseteq [n]$ can have $|\hat{f}(S)| \geq \theta$;

(ii) For any fixed $k \in [0,n]$, at most $1/\theta^2$ $S_i \subseteq \{1, \ldots, k\}$ have $\mathbb{E}[f_{k,S_1}(x)^2] \geq \theta^2$ (since $\sum_{S \subseteq [k]} \mathbb{E}[f_{k,S_1}(x)^2] = \sum_{S \subseteq [n]} \hat{f}(S)^2 = 1$);

(iii) If $k, S_1$ are such that $\mathbb{E}[f_{k,S_1}(x)^2] < \theta^2$ then $|\hat{f}(S)| < \theta \forall S$ such that $S \cap [k] = S_1$.

(indeed, $\mathbb{E}[f_{k,S_1}(x)^2] = \sum_{S' : S' \cap [k] = S_1} \hat{f}(S')^2 \geq \hat{f}(S)^2$ for any such $S$.)

Lemma 14. Fix $f : \{-1, 1\}^n \to \{-1, 1\}$, and pretend that $\mathbb{E}[f_{k,S_1}(x)^2]$ can be computed exactly in unit time given MQ($f$), $k \subseteq [k]$ and $x$. Then we can, given $\theta > 0$, exactly identify $\left\{ S \subseteq [n] : |\hat{f}(S)| \geq \theta \right\}$ in time $\text{poly}(n, \frac{1}{\theta})$.

The idea is to build a (partial) binary tree, where each level-$\ell$ node has “address” $(\ell, S'_\ell)$, and value $\sum_{T_2 \subseteq \{\ell+1, \ldots, n\}} \hat{f}(S'_\ell \cup T_2)^2 = \mathbb{E}[f_{\ell,S'_\ell}(x)^2]$. This tree (built on-the-fly by the algorithm exploring it) will have as key invariant that, for each $\ell$, $\sum_{S'_{\ell} \subseteq [\ell]} \text{value}(\ell, S'_{\ell}) \leq 1$. The algorithm will stop exploring $(\ell, S'_\ell)$ if $\text{value}(\ell, S'_\ell) \leq \theta^2$; therefore, for all $\ell$, the number of live nodes at level $\ell$ will be at
most $1/\theta^2$. This means the tree (which has depth at most $n$) can be entirely explored in time $\text{poly}(n, 1/\theta)$. Each leaf node that is reached at depth $n$ corresponds to a Fourier coefficient and this collection of leaves will contain all Fourier coefficients with $|\hat{f}(S)| \geq \theta$.

We give a formal proof of this in the next lecture.