1 Overview

Continuing from last lecture, we will finish presenting the learning algorithm of Linial, Mansour, and Nisan (LMN [1]). This will show that we can approximately learn constant-depth circuits (AC$_0$) in $O(n^{polylog(n)})$ time, given only oracle access. The algorithm will proceed by simply querying the function on random points, and estimating the low-degree Fourier coefficients from these points. For this to be approximately correct, we must show that low-depth circuits can be approximated by low-degree polynomials (i.e., their Fourier spectrum concentrates on low-degree terms).

The analysis will proceed roughly as follows:

1. Use Hastad’s Switching Lemma and Fourier analysis to show that low-depth circuits don’t have much mass in their higher-order Fourier coefficients.

2. Show that approximately learning a function is equivalent to approximately learning its Fourier coefficients.

3. Show that the probabilistic interpretation of Fourier coefficients (as correlations with parities) naturally gives a strategy for estimating the Fourier coefficients.

The bulk of the analysis is in Step 1. This is related to our previous discussion of circuit lower bounds, roughly because: if a circuit has large high-order Fourier coefficients, then it is essentially computing large parities (recall the Fourier basis is parities of subsets). But parities cannot be computed by circuits of low-depth, as we showed previously.

2 Notation

Let $F(x)$ be a binary function on $n$ bits:

$$F : \{-1, +1\}^n \rightarrow \{-1, +1\}$$

(1)

For sets $S \subseteq [n]$, let $\hat{\alpha}_S$ denote the Fourier coefficients of $F$. So $F$ can be written as the multilinear polynomial:

$$F(x) = \sum_{S \subseteq [n]} \hat{\alpha}_S \prod_{i \in S} x_i$$

(2)

Let $\chi_S$ denote the Fourier basis functions:

$$\chi_S(x) = \prod_{i \in S} x_i$$

(3)

So equivalently:

$$F(x) = \sum_{S \subseteq [n]} \hat{\alpha}_S \chi_S(x)$$

(4)
Recall that since the Fourier basis is orthonormal (with respect to the inner-product \( \langle f, g \rangle = \mathbb{E}_x[f(x)g(x)] \)), we have the simple Fourier inversion formula:

\[
\hat{\alpha}_S = \mathbb{E}_x[F(x)\chi_S(x)]
\] (5)

And since both \( F \) and \( \chi_S \) are \((\pm1)\)-valued, the correlation can be written in terms of their probability of agreeing on a random input:

\[
\hat{\alpha}_S = \mathbb{E}_x[F(x)\chi_S(x)] = \Pr_x[F(x) = \chi_S(x)] - \Pr_x[F(x) \neq \chi_S(x)]
\] (6)

3 Fourier Spectrum of Low-Depth Circuits

Here we show that low-depth circuits don’t have much mass in their higher-order Fourier coefficients. Instead of directly analyzing the Fourier representation of circuits, we will use Hastad’s Switching Lemma to consider a random restriction, which we know leads to a low-depth decision tree (and therefore has only small Fourier coefficients). We know how to relate the Fourier coefficients of a function before and after random restriction, so this will allow us to bound the Fourier spectrum of the original function.

Last lecture, we proved the following lemma:

**Lemma 1** If \( \rho \) is a random restriction of \( F \) leaving \( pn \) variables unset, then

\[
\mathbb{E} \left[ \sum_{T:|T| \geq D} (\hat{\alpha}'_T)^2 \right] \geq \Omega \left( \sum_{T:|T| \geq O \left( \frac{D}{p} \right)} (\hat{\alpha}_T)^2 \right)
\] (7)

Where \( \hat{\alpha}_T \) are Fourier coefficients of \( F \), and \( \hat{\alpha}'_T \) are the coefficients of \( F|_{\rho} \).

By the above lemma, it suffices to show that after random restriction, the higher-order Fourier coefficients are together very small w.h.p., so the original higher-order Fourier coefficients must also be small.

We will use the following lemma:

**Lemma 2** (Hastad’s Switching Lemma) If \( F \) has a depth \( d \), size \( s \) circuit (\( \lor, \land \) gates of unbounded fan-in), and \( D \geq \log s, p = O \left( \frac{1}{D^{d-1}} \right) \), then the random restriction \( F|_{\rho} \) has low decision-tree depth w.h.p.:

\[
\Pr_{\rho} \left[ F|_{\rho} \text{ has Decision-Tree depth } \geq D \right] \leq 2^{-D}
\] (8)

We showed last time that a depth-\( D \) decision tree can be represented (uniquely) by a degree-\( D \) multilinear polynomials, so in particular, a depth-\( D \) decision tree will have no Fourier coefficients of size \( > D \). Therefore:

**Corollary 1**

\[
\Pr_{\rho} [F|_{\rho} \text{ has any } \hat{\alpha}'_S \text{ s.t. } |S| > D] \leq 2^{-D}
\] (9)

Moreover:
Corollary 2
$$\mathbb{E}\left[\sum_{T:|T|\geq D} (\hat{a}'_T)^2\right] \leq 2^{-D} \quad (10)$$

Proof. In the $\leq 2^{-D}$ fraction of cases for which large coefficients exist (by Corollary 1), the sum $\sum_{T:|T|\geq D} (\hat{a}'_T)^2 \leq \sum T (\hat{\alpha}_T)^2 = 1$ by Parseval's identity. Combining these, we can bound the higher-order Fourier mass of the original function $F$:

Corollary 3
$$\sum_{T:|T|\geq O(D^d)} (\hat{a})^2 \leq O(2^{-D}) \quad (11)$$

Proof. Combine Lemma 1 with $p = O\left(\frac{1}{D^{D^d}}\right)$, and Corollary 2:

$$\Omega\left(\sum_{T:|T|\geq O(D^d)} (\hat{a}_T)^2\right) \leq \mathbb{E}\left[\sum_{T:|T|\geq D} (\hat{a}_T)^2\right] \leq O(2^{-D}) \quad (12)$$

Notice that we should expect that the higher-order Fourier coefficients of $F$ are individually small, but we have in fact shown the stronger result that they are collectively small. (Intuitively this should be true, because a function can't be simultaneously correlated to many large parities).

We have shown that low-depth circuits have only a small amount of mass in its high-order parities. But can we just ignore these coefficients in learning the function? Next we will show that we essentially can.

4 Approximate Learning in Fourier Domain

Here we will show that approximately learning a function is equivalent to approximately learning its Fourier coefficients. Thus, two functions whose Fourier coefficients largely agree will also largely agree as functions.

One issue is, once we start approximating Fourier coefficients, the resulting function may not take values strictly in $\{\pm 1\}$. Thus, in order to analyze this, we need to extend our boolean functions to $\mathbb{R}$. We have previously been writing functions as multilinear polynomials over the boolean hypercube, but we may simply extend them to be over $\mathbb{R}$.

Lemma 3 Let $f, g$ be multilinear polynomials over $\mathbb{R}$:

$$f = \sum_{S \subseteq \{n\}} \beta_S \prod_{i \in S} x_i, \quad g = \sum_{S \subseteq \{n\}} \gamma_S \prod_{i \in S} x_i$$

Then

$$\mathbb{E}_{x \in \{\pm 1\}^n} [(f(x) - g(x))^2] = \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} (f(x) - g(x))^2 = \sum_{S \subseteq \{n\}} (\beta_S - \gamma_S)^2 \quad (13)$$

Proof. This is clear, since the Fourier basis is orthogonal (with respect to the inner product $\mathbb{E}$), and orthogonal transforms preserve norm. ■
5 The Learning Algorithm

The main idea of the algorithm (and analysis) is: approximately learning a Fourier coefficient is easy, since Fourier coefficients are just correlations with parities (which can be approximately learnt by sampling). That is,

\[ \hat{\alpha}_S = \mathbb{E}_x[F(x)\chi_S(x)] = \Pr_x[F(x) = \chi_S(x)] - \Pr_x[F(x) \neq \chi_S(x)] \] (14)

The goal, given oracle access to some low-depth function \( F \), is to learn a hypothesis function \( h \) that agrees with \( F \) except perhaps on some \( \epsilon \)-fraction of inputs.

We will need to set some parameters. First, the number of low-degree Fourier coefficients (\( \hat{\alpha}_S \) with \( |S| \leq O(D^d) \)) that we are going to try to approximate is

\[ M = \left( \frac{n}{O(D^d)} \right) \] (15)

We will approximate each of these coefficients to within an additive factor of

\[ \delta = \left( \frac{\epsilon}{2M} \right)^{1/2} \] (16)

We will approximate the higher-order coefficients as 0. We want the Fourier mass in these remaining terms to be very small (inverse poly), so we will set

\[ D = O(\log s + \log(1/\epsilon)) \] (17)

so that \( \sum_{T : |T| \geq O(D^d)} (\hat{\alpha}_T)^2 \leq O(2^{-D}) \) is small (by Corollary 3).

**The algorithm is:**

1. Get about \( \frac{1}{\sqrt{\epsilon}} \log \left( \frac{M}{\epsilon} \right) \) random samples \((\vec{x}_i, f(\vec{x}_i))\) of the function.

2. For each small monomial \( \chi_T : |T| \leq O(D^d) \), use the samples to compute (using empirical probabilities):

\[ \gamma_T = \Pr_x[F(x) = \chi_T(x)] - \Pr_x[F(x) \neq \chi_T(x)] \] (18)

3. Compute a \( \mathbb{R} \)-valued approximation to \( F \) as:

\[ g(x) = \sum_{T : |T| \leq O(D^d)} \gamma_T \chi_T(x) \] (19)

4. Return the binary-valued hypothesis function

\[ h = \text{sign}(g) \] (20)

**Analysis**

We will show that that our hypothesis \( h \) agrees with \( F \) everywhere except perhaps on \( \epsilon \)-fraction of inputs:

\[ \Pr_x[F(x) \neq h(x)] \leq \epsilon \] (21)
First notice that $\sim 1/\delta^2$ samples is sufficient to approximate a single Fourier coefficient within an additive factor of $\delta$ (by Chernoff bound). The additional $\log(M)$ factor in Step 1 allows us to union-bound over all Fourier coefficients, and conclude that all our estimated Fourier coefficients (in Step 2) are approximately correct with high probability:

$$\forall T, |T| \leq O(D^d): \ |\gamma_T - \alpha_T| \leq \delta$$  \hspace{1cm} (w.h.p)

Then, estimating larger coefficients as 0, we can bound our total error in learning the Fourier coefficients:

$$\sum_S (\alpha_S - \gamma_S)^2 = \sum_{S: |S| \leq O(D^d)} (\alpha_S - \gamma_S)^2 + \sum_{S: |S| > O(D^d)} \alpha_S^2$$  \hspace{1cm} (22)

$$\leq \left( \sum_{S: |S| \leq O(D^d)} \delta^2 \right) + 2^{-D} \leq \epsilon$$  \hspace{1cm} (by choice of $\delta, D$)

Now by Lemma 3, the function $g$ (determined by our estimated Fourier coefficients $\gamma_S$) approximately agrees with the function $F$:

$$\mathbb{E}_x [(F(x) - g(x))^2] = \sum_S (\alpha_S - \gamma_S)^2 \leq \epsilon$$  \hspace{1cm} (23)

But $g$ is not binary, so we must bound the error when we take its sign. But this is easy:

$$\Pr_x [F(x) \neq h(x)] = \Pr_x [F(x) \neq \text{sign}(g(x))]$$  \hspace{1cm} (24)

$$= \mathbb{E}_x [\frac{1}{4} (F(x) - \text{sign}(g(x))^2)]$$  \hspace{1cm} (since both functions are $\pm 1$)

$$\leq \mathbb{E}_x [(F(x) - g(x))^2]$$  \hspace{1cm} (by considering the two cases $F(x) = \text{sign}(g(x))$ and $F(x) \neq ...$)

$$\leq \epsilon$$  \hspace{1cm} (25)

Therefore the hypothesis $h$ agrees with $F$ almost everywhere.

**Runtime**

The runtime of this algorithm is on the order of its sample complexity:

$$\frac{1}{\delta^2} \log \left( \frac{M}{\epsilon} \right) = \text{poly} \left( \frac{M}{\epsilon} \right)$$  \hspace{1cm} (26)

By our choice of $M = (\log(n))^d$, this is at most

$$n^{O(\log(s/\epsilon))^d}$$  \hspace{1cm} (27)

This concludes our analysis of LMN learning.

**References**