CS 294-114 – Fine-Grained Complexity and Algorithms	Sept 8, 2015
Lecture 4: LMN Learning (Part 2)	
Instructor: Russell Impagliazzo	*Scribe: Preetum Nakkiran*

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 (ie, 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:

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

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

$$F(x) = \sum_{S \subseteq [n]} \hat{\alpha}_S \prod_{i \in S} x_i \tag{2}$$

Let χ_S denote the Fourier basis functions:

$$\chi_S(x) = \prod_{i \in S} x_i \tag{3}$$

So equivalently:

$$F(x) = \sum_{S \subseteq [n]} \hat{\alpha}_S \chi_S(x) \tag{4}$$

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:

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

And since both F and χ_S are (± 1)-valued, the correlation can be written in terms of their probability of agreeing on a random input:

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

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 ρ is a random restriction of F leaving pn variables unset, then*

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

Where $\hat{\alpha}_T$ are Fourier coefficients of F, and $\hat{\alpha}'_T$ are the coefficients of $F\restriction_\rho$.

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 ($\vee, \wedge$ gates of unbounded fan-in), and $D \geq \log s$, $p = O(\frac{1}{D^{d-1}})$, then the random restriction $F\restriction_\rho$ has low decision-tree depth w.h.p.:*

$$\Pr_\rho\left[F\restriction_\rho \quad \textit{has Decision-Tree depth} \quad \geq D\right] \leq 2^{-D} \tag{8}$$

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

$$\Pr_\rho[F\restriction_\rho \quad \textit{has any} \quad \hat{\alpha}'_S \quad \textit{s.t.} \quad |S| > D] \leq 2^{-D} \tag{9}$$

Moreover:

Corollary 2

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

Proof. In the $\leq 2^{-D}$ fraction of cases for which large coefficients exist (by Corollary 1), the sum $\sum_{T:|T|\geq D} (\hat{\alpha}'_T)^2 \leq \sum_T (\hat{\alpha}'_T)^2 = 1$ by Parseval's identity. ■

Combining these, we can bound the higher-order Fourier mass of the original function F:

Corollary 3

$$\sum_{T:|T|\geq O(D^d)} (\hat{\alpha}_T)^2 \leq O(2^{-D}) \tag{11}$$

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

$$\Omega\left(\sum_{T:|T|\geq O(D^d)} (\hat{\alpha}_T)^2\right) \leq \mathbb{E}\left[\sum_{T:|T|\geq D} (\hat{\alpha}'_T)^2\right] \leq O(2^{-D}) \tag{12}$$

■

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 Fourier coefficients. 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

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

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,

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

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 ϵ-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

$$M = \binom{n}{O(D^d)} \tag{15}$$

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

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

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

$$D = O(\log s + \log(1/\epsilon)). \tag{17}$$

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 $\frac{1}{\delta^2}\log\left(\frac{M}{\epsilon}\right)$ 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):
$$\gamma_T = \Pr_x[F(x) = \chi_T(x)] - \Pr_x[F(x) \neq \chi_T(x)] \tag{18}$$
3. Compute a $\mathbb{R}$-valued approximation to F as:
$$g(x) = \sum_{T:|T|\leq O(D^d)} \gamma_T \chi_T(x) \tag{19}$$
4. Return the binary-valued hypothesis function
$$h = sign(g) \tag{20}$$

Analysis
We will show that that our hypothesis h agrees with F everywhere except perhaps on ϵ-fraction of inputs:

$$\Pr_x[F(x) \neq h(x)] \leq \epsilon \tag{21}$$

First notice that $\sim 1/\delta^2$ samples is sufficient to approximate a single Fourier coefficient within an additive factor of δ (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:

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

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

$$\begin{aligned}
\sum_S (\alpha_S - \gamma_S)^2 &= \sum_{S:|S|\leq O(D^d)} (\alpha_S - \gamma_S)^2 + \sum_{S:|S|>O(D^d)} \alpha_S^2 && (22)\\
&\leq = (\sum_{S:|S|\leq O(D^d)} \delta^2) + 2^{-D} \leq \epsilon && \text{(by choice of } \delta,\ D)
\end{aligned}$$

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

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

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

$$\begin{aligned}
\Pr_x[F(x) \neq h(x)] &= \Pr_x[F(x) \neq sign(g(x))] && (24)\\
&= \mathbb{E}_x[\frac{1}{4}\left(F(x) - sign(g(x))\right)^2] && \text{(since both functions are } \pm 1)\\
&\leq \mathbb{E}_x[(F(x) - g(x))^2] \\
&\qquad \text{(by considering the two cases } F(x) = sign(g(x)) \text{ and } F(x) \neq ...) \\
&\leq \epsilon && (25)
\end{aligned}$$

Therefore the hypothesis h agrees with F almost everywhere. ∎

Runtime

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

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

By our choice of $M = \binom{n}{O(D^d)}$, this is at most

$$n^{O(\log(s/\epsilon))^d} \tag{27}$$

This concludes our analysis of LMN learning.

References

[1] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.