STAT 206A: Polynomials of Random Variables	Learning Juntas
Lecture 9	
Lecture date: Sep 27	*Scribe: Kamalika Chaudhuri*

In the last class, we stated the following theorem about learnability of monotone functions.

Theorem 1 *In the PAC learning model, the class of all monotone functions* $f : \{-1,1\}^n \to \{-1,1\}$ *is learnable in time* $2^{O(\frac{1}{\epsilon}\sqrt{n}\log n)} \log(1/\delta)$.

This class we will prove the theorem and discuss some more results on learning functions of a small number of variables. The reference for all these results can be found in the paper [1]. Before proving the theorem, we need a couple of lemmas. The first lemma is about approximating real valued functions by their signs.

Lemma 2 *Let* (Ω, μ) *be a probability space and let* $f : \Omega \to \{-1,1\}$ *and* $g : \Omega \to \mathcal{R}$ *be two functions such that* $|f - g|_2^2 \leq \epsilon$. *If* $h = \mathrm{sgn}(g)$ *then*

$$|f - h|_2^2 \leq 4\epsilon$$

Proof: Let A denote the event that f and h disagree, viz.,

$$A = \{x \in \Omega : f(x) \neq h(x)\}$$

Then

$$|f - h|_2^2 = \mathbf{E}\left[(f-h)^2\right] = 4\mathbf{P}(A)$$

On the other hand,

$$|f - g|_2^2 \geq \mathbf{P}(A)$$

This is because whenever f and h disagree, g and f differ by at least 1, and this contributes at least 1 to the difference. □

The second lemma explains how to learn a function f which has the property that all but a small fraction of its fourier coefficients are concentrated in a set W.

Lemma 3 *Supppose we are given $W \subseteq 2^{[n]}$ and a function $f : \{-1,1\}^n \to \{-1,1\}$ such that*

$$\sum_{S \in W} \widehat{f}_S^2 \geq 1 - \epsilon$$

Then there exists an algorithm A which, given parameters $\delta > 0, \theta > 0$ runs in time polynomial in $|W|$, n, $1/\theta$ and $\log(1/\delta)$ and returns a function

$$g = \sum_{S \in W} c_S u_S$$

such that

$$\mathbf{E}\left[(f-g)^2\right] \leq \epsilon + \theta$$

except with probability δ.

Proof: We learn our function f as follows. We take a set of N samples, where N is to be specified later. We will show that N is a polynomial in $|W|$, n, $1/\theta$ and $\log(1/\delta)$. Using these N samples, we estimate each Fourier Coefficient $\widehat{f}_S$ for each S belonging to W. We claim that if $|\widehat{f}_S| > 1/n^c$ for some constant c, then $\widehat{f}_S$ is estimated correctly, that is within a factor of $(1+\lambda)$ of $\widehat{f}_S$. This holds because we can use Chernoff Bounds to say that

$$\begin{aligned} \mathbf{P}[|c_S - \widehat{f}_S| \geq \lambda \widehat{f}_S] &\leq \exp\left(-N\lambda^2 |\frac{1}{2} - \widehat{f}_S|\right) \\ &\leq \exp\left(-N\lambda^2/n^c\right) \end{aligned}$$

Using the Union Bound, the probability that all the coefficients in W are estimated correctly is at most $|W| \exp\left(-N\lambda^2/n^c\right)$. We want this probability to be at most δ. To ensure this, we set

$$N = \frac{n^c}{\lambda^2} \log\left(\frac{|W|}{\delta}\right) \tag{1}$$

Now we will estimate the value of λ needed to ensure the error guarantees.

$$\begin{aligned} \mathbf{E}\left[(f-g)^2\right] &= \mathbf{E}\left[(\sum_S (\widehat{f}_S - c_S) u_S(x))^2\right] \\ &= \mathbf{E}\left[(\sum_{S \in W} (\widehat{f}_S - c_S) u_S(x) + \sum_{S \notin W} \widehat{f}_S u_S(x))^2\right] \\ &\leq \epsilon + |W| \max_{S \in W} |f_S - c_S| \\ &\leq \epsilon + |W| \max(\lambda, 1/n^c) \end{aligned}$$

The second expression is equal to θ for $\lambda = \theta/|W|$ (assuming that $1/n^c$ is much smaller than θ). Plugging in to Equation 1, the total number of samples needed is at most $\frac{|W|^2 n^c}{\theta^2} \log\left(\frac{|W|}{\delta}\right)$. $\square$

Now we are ready to prove Theorem 1.

Proof:(Of Theorem 1) We know that all monotone functions f on $\{-1,1\}^n$ satisfy the following bound on the total influence of the variables.

$$\sum_{i=1}^{n} I_i(f) \leq c\sqrt{n} \tag{2}$$

where c is a constant. Since the total influence can also be written as $\sum_S |S|\widehat{f}_S^2$, this means that for all $\epsilon > 0$,

$$\sum_{|S| > \frac{c\sqrt{n}}{\epsilon}} \widehat{f}_S^2 \leq \epsilon$$

Now if we pick W to be the set of all subsets of $[n]$ with size at most $c\sqrt{n}$, and use the algorithm described in the previous lemma, the theorem will follow. This is because the size of W is $\binom{n}{\frac{c\sqrt{n}}{\epsilon}}$ which is at most $2^{O(\frac{\sqrt{n}\log n}{\epsilon})}$. $\square$

Definition 4 *We define C_n^k to be the class of all boolean functions from $\{-1,1\}^n \to \{-1,1\}$ which depend on only k coordinates.*

For the rest of the class, we will show a few lemmas on how to learn the functions in C_n^k with a small number of samples. We will eventually show that C_n^k is learnable in time $n^{\alpha k + \theta(1)} \log(1/\delta)$ where $\alpha = \frac{\omega}{1+\omega}$, where $\omega = 2.37$, the matrix multiplication constant.

Lemma 5 *Suppose we have an algorithm A, which, when given a function $f \in C_n^k$, outputs one of the variables with nonzero influence in time $C(k)n^{\gamma_k} \log(1\ \delta)$. Then there is an algorithm A' which learns C_n^k in time $C'(k)n^{\gamma_k} \log(1/\delta)$.*

Proof: Algorithm A' works by first running A to find a variable with nonzero influence. Once such a variable is found, we fix its value, and run A' again with the variable held constant. This outputs another variable. Proceeding in this manner, we can obtain all the influential variables by running A at most 2^k times. $\square$

Definition 6 *For $1 \leq r \leq k$ we define $C_n^k(r)$ to be the subclass of functions in C_n^k for which $\widehat{f}_S \neq 0$, for some set $S \neq \emptyset$ of size at most r. We let $C_n^k(0)$ be the class of non-balanced functions.*

Lemma 7 $C_n^k(0) \setminus \{-1,1\} \subseteq C_n^k\left(\lceil \frac{2k}{3} \rceil\right)$

Proof: Let $r = \lceil \frac{2k}{3} \rceil$. Suppose for the sake of contradiction,

$$f = a_0 + \sum_{|S|>r} a_S u_S \tag{3}$$

where $u_S = \prod_{j \in S} x_j$ for $a_0 \neq 0$. Then

$$\begin{aligned} 1 = f^2 &= \left(a_0 + \sum_{|S|>r} a_S u_S \right)^2 \\ &= a_0^2 + \sum_{|S|>r} a_S^2 + 2a_0 \sum_{|S|>r} a_S u_S + \sum_{|S|,|S'|>r} a_S a_{S'} u_S u_{S'} \\ &= a_0^2 + \sum_{|S|>r} a_S^2 + 2a_0 \sum_{|S|>r} a_S u_S + \sum_{|S|,|S'|>r} a_S a_{S'} u_{S\Delta S'} \end{aligned}$$

The last two terms must cancel as 1 is a constant. Note that the last term is actually a weighted sum of Walsh functions for sets of size strictly less than r, since $|S \Delta S'| \leq 2(k-r) < r$, and the second last term is a weighted sum of Walsh functions for sets of size r or more. Hence they cannot cancel unless f is a constant function. $\square$

Exercise 8 *(1 pt) Is this lemma tight?*

Note that there is an algorithm which, given a function f belonging to the class $C_n^k(r)$, outputs a variable which has nonzero influence in time $C(k)n^r \log(1/\delta)$. This can be easily done by estimating the Fourier Coefficients of all sets of size at most r, and outputting a variable j from a set S whose estimated Fourier Coefficient is at least $\frac{1}{2^k}$ awayfrom zero. The number of samples needed for this to succeed with probability at least $1-\delta$ is $C(k) \log n \log(1/\delta)$.

References

[1] E. Mossel, R. O'Donnell, and R. A. Servedio. Learning juntas. In *Proceedings of the 35th Annual symposium on the theory of computing (STOC)*, pages 206–212, 2003.