Computer Science 294 Lecture 7 Notes

Daniel Raban

February 7, 2023

1 Low Degree Learning and Goldreich-Levin's Algorithm

1.1 Recap: weights and approximation of boolean functions

Recall that if we have a boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, then we can write

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \prod_{i \in S} x_i.$$

We had Parseval's identity

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = 1$$

and defined the weights at different degrees as

$$W^k(f) := \sum_{S : |S| = k} \widehat{f}(S)^2, \qquad W^{>k} := \sum_{S : |S| > k} \widehat{f}(S)^2.$$

We said that f is **ε-concentrated up to degree** k if $W^{>k} \leq \varepsilon$. We also saw that f is well-concentrated up to degree k if and only if f is well-approximated in ℓ_2-norm by $\deg k$ polynomials.

1.2 PAC learning

Today we will be talking about PAC (probably approximately correct) learning [Valiant '84]. The motivation is that given many examples, we want to learn a "simple" hypothesis that explains the data and generalizes.

We make the assumption that the data itself is labeled according to a "simple" function like

- k-junta
- low depth decision tree

- small size decision tree.

More formally, suppose you have a **concept class** $\mathcal{C} \subseteq \{f : \{\pm 1\}^n \to \{\pm 1\}\}$, for example decision trees. Let $f \in \mathcal{C}$ be unknown to you. You get a collection of random labeled examples $(x^{(1)}, f(x^{(1)})), (x^{(2)}, f(x^{(2)})), \dots$ where each $x^{(i)}$ is selected uniformly at random from $\{\pm 1\}^n$. The goal is to output a hypothesis $h : \{\pm 1\}^n \to \{\pm 1\}^n$ such that with probability at least $1 - \delta$, the hypothesis is ε-close to f. That is,

$$\mathbb{P}_{X \sim \{\pm 1\}^n}(h(X) \neq f(X)) \leq \varepsilon.$$

Valiant originally considered this for distributions which were not necessarily uniform. In that case, you need to compare h and f with respect to that distribution. We will only focus on the uniform case.

Theorem 1.1 (Linial-Mansour-Nisan)**.** *Suppose $\mathcal{C}$ is a concept class such that any $f \in \mathcal{C}$ is ε-concentrated up to degree k. Then $\mathcal{C}$ is PAC-learnable (over the uniform distribution) in time* $\mathrm{poly}(n^k, 1/\varepsilon, \log(1/\delta))$.

We will show that with probability $\geq 1 - \delta$, the algorithm would output h such that

$$\mathbb{P}_{X \sim \{\pm 1\}^n}(h(X) \neq f(X)) \leq 2\varepsilon.$$

Before proving this theorem, we will first prove a lemma:

Lemma 1.1. *Let $f : \{\pm 1\}^n \to \{\pm 1\}$ and $S \subseteq [n]$. Then, given random labeled examples, we can estimate $\widehat{f}(S)$ up to additive accuracy ε, with probability at least $1 - \delta$ in time $O(n \cdot \log(1/\delta)/\varepsilon^2)$.*

This is a direct consequence of Hoeffding's inequality.

Lemma 1.2 (Chernoff-Hoeffding)**.** *If $Z_1, \dots, Z_n$ are iid and bounded ($-1 \leq Z_i \leq 1$), then*

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mathbb{E}[Z_1]\right|\right) \leq 2e^{-\varepsilon^2 m/2}.$$

Proof. Recall that $\widehat{f}(S) = \mathbb{E}_{X \sim \{\pm 1\}^n}[f(X)\chi_S(X)]$. Sample m inputs uniformly at random: $x^{(1)}, x^{(2)}, \dots, x^{(m)}$, and calculate the empirical mean $\widetilde{f}(S) = \frac{1}{m}\sum_{i=1}^{m} f(x^{(i)})\chi_S(x^{(i)})$. Then, by Chernoff with $Z_i = f(x^{(i)})\chi_S(x^{(i)})$ (so $\mathbb{E}[Z_1] = \widehat{f}(S)$,

$$\mathbb{P}(|\widetilde{f}(S) - \widehat{f}(S)| \geq \varepsilon) \leq 2e^{-\varepsilon^2 m/2}.$$

If we pick $m = \frac{2}{\varepsilon^2} \cdot \log(2/\delta)$, this is $\leq \delta$. $\square$

Now we'll prove the theorem.

Proof. Here is the algorithm:

1. For every set $S \subseteq [n]$ of size $\leq k$, estimate $\widehat{f}(S)$ up to accuracy $\varepsilon' = \sqrt{\varepsilon/n^k}$ and failure probability $\delta' = \delta/n^k$. This gives us the estimates $\widetilde{f}(S)$.
2. Output $h(x) = \operatorname{sgn}(\sum_{|S| \leq k} \widetilde{f}(S) \prod_{i \in S} x_i)$.

By the lemma, using a union bound, with probability $\geq 1 - \delta$, all the estimates $\widetilde{f}(S)$ are ε'-close to $\widehat{f}(S)$. Let's call this event "the good case." In this case, let $p(x) = \sum_{|S| \leq k} \widetilde{f}(S) \prod_{i \in S} x_i$, so $h(x) = \operatorname{sgn}(p(x))$. Then

$$\mathbb{P}_{X \sim \{\pm 1\}^n}(f(X) \neq h(X)) = \mathbb{P}_X(f(X) \neq \operatorname{sgn}(p(X)))$$

Since f is $\{\pm 1\}$-valued, if $f(x) \neq \operatorname{sgn}(p(x))$, then $|f(x) - p(x)| \geq 1$. So we can bound this probability by an ℓ_2 distance.

$$\begin{aligned}
&\leq \mathbb{E}_{X \sim \{\pm 1\}^n}[(f(X) - p(X))^2] \\
&= \sum_{S \subseteq [n]} (\widehat{f}(X) - p(S))^2 \\
&= \sum_{|S|:|S| \leq k} (\widehat{f}(S) - \widetilde{f}(S))^2 + \sum_{S:|S|>k} \widehat{f}(S)^2 \\
&\leq (\varepsilon')^2 \cdot \left(1 + n + \binom{n}{2} + \cdots + \binom{n}{k}\right) + \varepsilon \\
&= \underbrace{\frac{\varepsilon}{n^k}\left(1 + n + \binom{n}{2} + \cdots + \binom{n}{k}\right)}_{\leq \varepsilon} + \varepsilon \\
&\leq 2\varepsilon.
\end{aligned}$$

□

Corollary 1.1. *Depth-d decision trees are PAC learnable (over the uniform distribution) in time $n^{O(d)}$.*

Corollary 1.2. *Size-s decision trees are PAC learnable (over the uniform distribution) in time $n^{O(\log s)}$.*

Corollary 1.3. *LTFs (weighted majorities) can be learned in time $n^{O(1/\varepsilon^2)}$.*

Remark 1.1. This algorithm won't give you a decision tree, necessarily, but it will give a boolean function that approximates the decision tree.

It is open whether there are much better algorithms for learning depth-d decision trees or size-s decision trees in $\operatorname{poly}(s, n)$ time. Even the easier question of if we can learn $\log(n)$-juntas in $\operatorname{poly}(n)$ time is open.

1.3 Goldreich-Levin's Algorithm

In cryptography, a **one-way permutation (OWP)** is a permutation $f : \{\pm 1\}^n \to \{\pm 1\}^n$ which is "easy to compute" but "hard to invert." If $m < n$, another cryptographic primitive is a **pseudorandom generator (PRG)**, a function $G : \{\pm 1\}^m \to \{\pm 1\}^n$ where $G(U_m)$ is indistinguishable from U_n; essentially we want to take m random bits and create n random bits which seem uniformly distributed to any algorithm.

Given a OWP $f : \{\pm 1\}^n \to \{\pm 1\}^n$, let $G : \{\pm 1\}^{2n} \to \{\pm 1\}^{2n+1}$ be

$$g(r, s) = (r, f(s), \mathrm{IP}_2(r, s)),$$

where IP_2 is the inner product mod 2, viewing the inputs as elements of $\mathbb{F}_2$. As an exercise, show that if f is a OWP, then G is a PRG.

Goldreich-Levin is actually a learning algorithm in the membership query model. The setting is the same as in PAC learning, but the learner can request/query the value of $f(x)$ for any $x \in \{\pm\}^n$.

Theorem 1.2 (Goldreich-Levin)**.** *Given query access to f, there exists an algorithm that finds all "heavy" Fourier coefficients of f. Namely, given $\theta \in (0, 1)$, the algorithm outputs with high probability a list $\mathcal{L}$ such that*

$$|\widehat{f}(S)| \geq \theta \implies S \in \mathcal{L}, \qquad S \in \mathcal{L} \implies |\widehat{f}(S)| \geq \theta/2.$$

The algorithm's runtime is $n \operatorname{poly}(1/\theta)$.

These conditions imply that the list $\mathcal{L}$ will have $\leq 4/\theta^2$ elements. Here is how we connect this theorem back to learning theory.

Corollary 1.4 (Kushilevitz-Mansour)**.** *Let $\mathcal{C}$ be a concept class such that any $f : \{\pm 1\}^n \to \{\pm 1\}$ in $\mathcal{C}$ is ε-concentrated on at most M (unknown) coefficients. Then, $\mathcal{C}$ is learnable using queries with accuracy $O(\varepsilon)$ in time $\operatorname{poly}(M, n, 1/\varepsilon)$.*

Here are some consequences.

Corollary 1.5. *Decision trees of depth d are learnable with queries in $\operatorname{poly}(n) \cdot 2^{O(d)}$ time.*

Corollary 1.6. *Decision trees of size s are learnable with queries in $\operatorname{poly}(s, n)$ time.*

Corollary 1.7. *k-juntas are learnable with queries in $\operatorname{poly}(n) \cdot 2^{O(k)}$ time.*

Let's first show how the Goldreich-Levin theorem implies the corollary. We will prove the Goldreich-Levin next time.

Proof. Take $\theta = \sqrt{\varepsilon/M}$, and apply Goldreich-Levin's algorithm to get the list $\mathcal{L}$. Output a hypothesis h by running the LMN algorithm on $\mathcal{L}$; we can use this algorithm for any arbitrary collection of Fourier coefficients, not just the ones for sets of size $\leq k$. By assumption on f, there exists a set $\mathcal{F}$ of size M such that

$$\sum_{S \in \mathcal{F}} \widehat{f}(S)^2 \leq \varepsilon, \qquad \sum_{S \in \mathcal{F}} \widehat{f}(S)^2 \geq 1 - \varepsilon.$$

Assuming GL gave the list as guaranteed with high probability. We know that

$$\mathcal{L} \supseteq \{S : |\widehat{f}(S)| \geq \theta\}, \qquad |\mathcal{L}| \leq \frac{4}{\theta^2}.$$

Now look at

$$\begin{aligned}
\sum_{S \notin \mathcal{L}} \widehat{f}(S)^2 &= \sum_{S \notin \mathcal{L}, S \in \mathcal{F}} \widehat{f}(S)^2 + \sum_{S \notin \mathcal{L}, S \notin \mathcal{F}} \widehat{f}(S)^2 \\
&\leq M\theta^2 + \varepsilon \\
&\leq M \left(\sqrt{\frac{\varepsilon}{M}} \right)^2 + \varepsilon \\
&= 2\varepsilon.
\end{aligned}$$

Now the LMN algorithm gives a hypothesis that is $O(\varepsilon)$-close to f. $\square$

Here is the idea of the GL algorithm: For $f : \{\pm 1\}^n \to \{\pm 1\}$, we want to find the set of coefficients $\{S : |\widehat{f}(S)| \geq \theta\}$. The idea is to look at all sets, so $\sum \widehat{f}(S)^2 = 1$. Now split into two cases: all sets that include 1 and all sets that do not include 1. We will show that we can calculate $\sum \widehat{f}(S)^2$ in each case and recursively look at the sets which do or do not contain the next element.