6.842 Randomness and Computation April 8, 2022

Lectures on Learning Theory

Lecturer: Ronitt Rubinfield *Scribe: Yuchong Pan*

1 PAC Learning

The model of *learning from random, uniform examples* is as follows: Given the *example oracle* $\mathrm{Ex}(f)$ of a function f, pick m i.i.d. random variables $x_1, \ldots, x_m$ uniformly (or from some distribution $\mathcal{D}$, which might not be known to the learner in general), and call $\mathrm{Ex}(f)$ to obtain m random labeled examples $(x_1, f(x_1)), \ldots, (x_m, f(x_m))$; after seeing these examples, the learner outputs a hypothesis h of the function f.

Should we require $h = f$? This is probably too much to ask. However, we can at least require $\mathrm{dist}(h, f) := \mathbb{P}_{x\sim\mathcal{D}}(h(x) \neq f(x)) \leq \varepsilon$, where $\mathrm{dist}(h, f)$ is also called $\mathrm{error}_{\mathcal{D}}(h)$ with respect to f.

Definition 1. A *uniform distribution learning algorithm for a concept class* $\mathcal{C}$ is an algorithm $\mathcal{A}$ that, given $\varepsilon > 0$, $\delta > 0$ and access to $\mathrm{Ex}(f)$ for $f \in \mathcal{C}$, outputs a function h such that with probability at least $1 - \delta$, $\mathrm{error}(h)$ with respect to f is at most ε. This is called *probably approximately correct (PAC) learning.*

We are interested in the following parameters:

- m, the *sample complexity*;
- ε, the *accuracy* parameter;
- δ, the *confidence* parameter;
- the running time, which we hope to be $\mathrm{poly}(\log(\text{domain size}), 1/\varepsilon, 1/\delta)$;
- the *description* of h, which at least should be compact (i.e., $O(\log |\mathcal{C}|)$) and efficient to evaluate; it require $h \in \mathcal{C}$, then this is called *proper learning.*

Note that the uniform case is a special case of the PAC model. The more general PAC model is given $\mathrm{Ex}_{\mathcal{D}}(f)$ and bounds $\mathrm{error}_{\mathcal{D}}(h)$ with respect to f.

2 Learning Conjunctions

Let $\mathcal{C}$ be the class of conjunctions (i.e., 1-term DNF) over $\{0,1\}^n$. We cannot hope for 0-error from a sub-exponential number of random samples; to see this, note that it is hard to distinguish $f(x) = x_1 \cdots x_n$ and $f(x) = \mathsf{F}$. Algorithm 1 gives a polynomial time sampling algorithm for conjunction learning, where "**?**" indicates a parameter to be determined.

For x_i in the conjunction, it must be set in the same way in each positive example, so $i \in V$. For x_i not in the conjunction,

$$\mathbb{P}[i \in V] = \mathbb{P}\left[x_i \text{ is set is the same way in each of the } k \text{ positive examples}\right] = \frac{1}{2^{k-1}}.$$

By the union bound,

$$\mathbb{P}\left[\text{any } x_i \text{ not in the conjunction survives}\right] \leq \frac{n}{2^{k-1}} \leq \delta,$$

draw poly$(1/\varepsilon)$ samples
estimate $\mathbb{P}[f(x) = 1]$ to additive error at most $\pm\varepsilon/4$ and confidence at least $1 - \delta/2$
if *estimate is less than* $\varepsilon/2$ **then**
    return $h(x) = 0$
(estimate is at least $\varepsilon/2$; see a new positive example every $O(1/\varepsilon)$ samples)
collect ? more positive examples
$V \leftarrow$ set of indices of variables that are set in the same way in each positive example
return $h(x) = \bigwedge_{i \in V} x_i^{b_i}$, *where each* b_i *indicates if* x_i *is complemented or not*

Algorithm 1: A polynomial time sampling algorithm for conjunction learning.

if we pick $k = \log(n/\delta)$. Therefore, if we need $\Omega(\log(n/\delta))$ positive examples, or $\Omega((1/\varepsilon)\log(n/\delta))$ total examples to rule out every x_i not in the conjunction.

3 Occam's Razor

In a high level, *Occam's Razor* claims the following:

- If we ignore the running time, then learning is easy (with a polynomial number of samples).
- The shortest explanation is the best.

To see the first claim, we consider the brute-force algorithm in Algorithm 2.

draw $M = (1/\varepsilon)(\ln|\mathcal{C}| + \ln|1/\delta|)$
search over all $h \in \mathcal{C}$ until find one consistent with the samples
return h

Algorithm 2: A brute-force learning algorithm that demonstrates Occam's Razor.

We say that a function h is *bad* if error(h) with respect to f is at least ε. For a bad function h,

$$\mathbb{P}[h \text{ is consistent with the samples}] \leq (1-\varepsilon)^M.$$

By the union bound,

$$\mathbb{P}[\text{any bad function } h \text{ is consistent with the samples}] \leq |\mathcal{C}|(1-\varepsilon)^M = |\mathcal{C}|(1-\varepsilon)^{\frac{1}{\varepsilon}\left(\ln|\mathcal{C}| + \ln\left|\frac{1}{\delta}\right|\right)} = \delta.$$

Hence, it is unlikely to output a bad hypothesis h. For example, for conjunction learning, this analysis requires $O((1/\varepsilon)(n + 1/\delta))$ samples, where Algorithm 1 has a better sample complexity. On the other hand, if we have a *good* hypothesis h,

(i) we can *predict* values of f on new random inputs according to distribution $\mathcal{D}$, since

$$\mathbb{P}_{x \sim \mathcal{D}}[f(x) = h(x)] \geq 1 - \delta;$$

(ii) we can *compress* the description of samples $(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_m, f(x_m))$ from the naïve description which takes $m(\log|D| + \log|R|)$ bits, where D and R are the domain and the range of f, respectively, to the form "$x_1, \ldots, x_m$ plus the description of h" which requires $m \log|D| + \log|\mathcal{C}|$ bits only.

4 Learning via Fourier Representations

In this section, we study learning algorithms that are based on estimating the Fourier representation of a function f.

4.1 Approximating One Fourier Coefficient

Lemma 2. *For any $S \subset [n]$, one can approximate $\hat{f}(S)$ to within additive error γ (i.e., $|output - \hat{f}(S)| \leq \gamma$) with probability at least $1 - \delta$ in $O(1/\gamma^2 \log 1/\delta)$ samples.*

Proof. Recall that $\hat{f}(S) = 2\,\mathbb{P}_{x \in \{0,1\}^n}[f(x) \neq \chi_S(x)] - 1$. Hence, we can approximate $\hat{f}(S)$ by estimating $\mathbb{P}_{x \in \{0,1\}^n}[f(x) \neq \chi_S(x)]$ and applying the Chernoff bound. □

4.2 Fourier Concentration and the Low Degree Algorithm

Definition 3. For all $\varepsilon \in (0,1)$ and $n \in \mathbb{N}$, we say that a function $f : \{\pm 1\}^n \to \mathbb{R}$ has $\alpha(\varepsilon, n)$-*Fourier concentration (f.c.)* if

$$\sum_{\substack{S \subset [n] \\ |S| > \alpha(\varepsilon,n)}} \hat{f}(S)^2 \leq \varepsilon.$$

For a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, Parseval's identity gives $\sum_{S \subset [n]} \hat{f}(S)^2 = 1$, so having $\alpha(\varepsilon, n)$-Fourier concentration implies that

$$\sum_{\substack{S \subset [n] \\ |S| \leq \alpha(\varepsilon,n)}} \geq 1 - \varepsilon.$$

The *low degree algorithm*, given in Algorithm 3, approximates a Boolean function with d-Fourier concentration, where $d = \alpha(\varepsilon, n)$.

```
take m = O((n^d/τ) log(n^d/δ)) samples
foreach S ⊂ [n] such that |S| ≤ d do
    C_S ← estimate of f̂(S)
let h : {±1}^n → ℝ be defined by h(x) = Σ_{S⊂[n]:|S|≤d} C_S χ_S(x)
return sign ∘h as hypothesis
```

Algorithm 3: The low degree algorithm given degree d, accuracy τ and confidence δ.

Proposition 4. *If f has d-Fourier concentration with $d = \alpha(\varepsilon, n)$, then $\mathbb{E}_{x \in \{0,1\}^n}[(f(x) - h(x))^2] \leq \varepsilon + \tau$ with probability at least $1 - \delta$.*

Proof. First, we claim that each low degree Fourier Coefficient is well approximated, i.e., with probability at least $1 - \delta$, we have $|C_S - \hat{f}(S)| \leq \gamma$ for all $S \subset [n]$ with $|S| \leq d$, where $\gamma = \sqrt{\tau/n^d}$. This can be proved using the Chernoff bound and the union bound.

Second, we show that if all low degree Fourier coefficients are well approximated, then h has a low ℓ_2-error. Suppose $|C_S - \hat{f}(S)| \leq \gamma$ for all $S \subset [n]$ such that $|S| \leq d$. Let $g : \{\pm 1\}^n \to \mathbb{R}$ be defined by

$$g(x) = f(x) - h(x).$$

By the linearity of the Fourier transform, for all $S \subset [n]$,

$$\hat{g}(S) = \hat{f}(S) - \hat{h}(S).$$

For all $S \subset [n]$ with $|S| > d$, we have $\hat{h}(S) = 0$, so $\hat{g}(S) = \hat{f}(S)$. For all $S \subset [n]$ with $|S| \le d$, we have $|\hat{g}(S)| \le \gamma$, so $\hat{g}(S)^2 \le \gamma^2$. Therefore,

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\in\{\pm1\}^n}\left[(f(x)-h(x))^2\right] &= \mathbb{E}\left[g(x)^2\right] = \sum_{S\subset[n]} \hat{g}(S)^2 && \text{(Parseval's identity)}\\
&= \sum_{\substack{S\subset[n]\\|S|\le d}} \hat{g}(S)^2 + \sum_{\substack{S\subset[n]\\|S|> d}} \hat{g}(S)^2\\
&\le \sum_{\substack{S\subset[n]\\|S|\le d}} \gamma^2 + \sum_{\substack{S\subset[n]\\|S|> d}} \hat{f}(S)^2\\
&\le \binom{n}{d}\cdot\gamma^2 + \varepsilon && \text{(by Fourier concentration)}\\
&\le \tau + \varepsilon.
\end{aligned}$$

This completes the proof. □

Proposition 5. *Let $f : \{\pm1\}^n \to \{\pm1\}$. Let $h : \{\pm1\}^n \to \mathbb{R}$. Then*

$$\mathop{\mathbb{P}}_{x\in\{\pm1\}^n}[f(x) \ne \operatorname{sign}(h(x))] \le \mathop{\mathbb{E}}_{x\in\{\pm1\}^n}\left[(f(x)-h(x))^2\right]$$

Proof. Recall that

$$\mathop{\mathbb{E}}_{x\in\{\pm1\}^n}\left[(f(x)-h(x))^2\right] = \frac{1}{2^n}\sum_{x\in\{\pm1\}^n}(f(x)-h(x))^2, \tag{1}$$

$$\mathop{\mathbb{P}}_{x\in\{\pm1\}^n}[f(x) \ne \operatorname{sign}(h(x))] = \frac{1}{2^n}\sum_{x\in\{\pm1\}^n}\mathbb{1}_{f(x)\ne\operatorname{sign}(h(x))}. \tag{2}$$

We compare (1) and (2) term by term. Let $x \in \{\pm1\}^n$. If $f(x) = \operatorname{sign}(h(x))$, then $\mathbb{1}_{f(x)\ne\operatorname{sign}(h(x))} = 0 \le (f(x)-h(x))^2$. If $f(x) \ne \operatorname{sign}(h(x))$, then $\mathbb{1}_{f(x)\ne\operatorname{sign}(h(x))} = 1 \le (f(x)-h(x))^2$; see Figure 1 for an illustration.

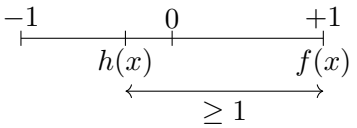

Figure 1: Illustrating the proof of Proposition 5.

□

Theorem 6. *If a concept class $\mathcal{C}$ has Fourier concentration $d = \alpha(\varepsilon, n)$, then there exists a uniform distribution learning algorithm for $\mathcal{C}$ with $q = O((n^d/\varepsilon)\log(n^d/\delta))$ samples; i.e., this algorithm gets q samples and with probability at least $1-\delta$ outputs a hypothesis h' such that*

$$\mathop{\mathbb{P}}_{x\in\{\pm1\}^n}\left[f(x) \ne h'(x)\right] \le 2\varepsilon.$$

Proof. Run the low degree algorithm with $\tau = \varepsilon$. By Proposition 4, we get a hypothesis h such that $\mathbb{E}_{x\in\{\pm1\}^n}[(f(x) - h(x))^2] \le \varepsilon + \varepsilon = 2\varepsilon$. Let $h' = \operatorname{sign} \circ h$. By Proposition 5, h' has error at most 2ε with respect to f. This completes the proof. □

Following are examples of functions that have $\alpha(\varepsilon, n)$-Fourier concentration.

(i) Any function $f : \{\pm1\}^n \to \mathbb{R}$ that depends on at most k variables has

$$\sum_{\substack{S\subset[n]\\|S|>k}} \hat{f}(S)^2 = 0.$$

(ii) Let $T = \{i_1, \dots, i_k\} \subset [n]$ be such that $|T| = k$. Let $\mathsf{AND} : \{\pm1\}^n \to \{\pm1\}$ be defined by

$$\mathsf{AND}(x) = \begin{cases} -1, & \text{if } x_i = -1 \text{ for all } i \in T, \\ 1, & \text{otherwise.} \end{cases}$$

We claim that AND has $\log(4/\varepsilon)$-Fourier concentration. Note $\widehat{\mathsf{AND}}(S) = 0$ for all $S \subset [n]$ with $|S| > |T|$. If $|T| \le \log(4/\varepsilon)$, then we are done by definition. If $|T| > \log(4/\varepsilon)$, then

$$\widehat{\mathsf{AND}}(\emptyset)^2 = (1 - 2\,\mathbb{P}\,[f(x) \neq \chi_\emptyset(x)])^2 = \left(1 - 2 \cdot \frac{1}{2^{|T|}}\right)^2 > 1 - \varepsilon.$$

Therefore, AND has 0-Fourier concentration.

(iii) Let $T = \{i_1, \dots, i_k\} \subset [n]$ be such that $|T| = k$. Let $\overline{\mathsf{AND}} : \{\pm1\}^n \to \{\pm1\}$ be defined by

$$\overline{\mathsf{AND}}(x) = \begin{cases} 1, & \text{if } x_i = -1 \text{ for all } i \in T, \\ -1, & \text{otherwise.} \end{cases}$$

Let $f : \{\pm1\}^n \to \{0, 1\}$ be defined by

$$\begin{aligned} f(x) &= \begin{cases} 1, & \text{if } x_i = -1 \text{ for all } i \in T, \\ 0, & \text{otherwise,} \end{cases} \\ &= \frac{1 - x_{i_1}}{2} \cdot \frac{1 - x_{i_2}}{2} \cdots \frac{1 - x_{i_k}}{2} \\ &= \frac{1}{2^k} \sum_{S\subset T} (-1)^{|S|} \chi_S(x). \end{aligned}$$

Note that all Fourier coefficients $\hat{f}(S)$ for $S \not\subset T$ are 0. Then

$$\overline{\mathsf{AND}}(x) = 2f(x) - 1 = -1 + \frac{2}{2^k} + \sum_{\substack{S\subset T\\S\neq\emptyset}} \frac{(-1)^{|S|}}{2^k} \chi_S(x).$$

(iv) ***Decision trees.*** Consider a decision tree T, e.g., Figure 2. For each leaf ℓ of T, let V_ℓ denote the set of indices of variables visited on the path from the root to leaf ℓ, and let $f_\ell : \{\pm1\}^n \to \{0, 1\}$ be defined by

$$f_\ell(x) = \prod_{i\in V_\ell} \frac{1 \pm x_i}{2} \qquad \text{(“±” denotes a left turn or a right turn)}$$

$$= \frac{1}{2^{|V_\ell|}} \sum_{S \subset V_\ell} (-1)^{\#\text{ left turns taken in } S} \chi_S.$$

Let $f : \{\pm 1\}^n \to \{\pm 1\}$ be defined by

$$f(x) = \sum_{\ell\text{: leaf of } T} f_\ell(x)\,\mathrm{val}(\ell).$$

Note that for each $x \in \{\pm 1\}^n$, exactly one of the values $f_\ell(x)$ is 1 for leaves ℓ of T, and all others are 0. Moreover, for each leaf ℓ of T, the number of variables on which f_ℓ depends is at most the depth of ℓ. By the linearity of the Fourier transform,

$$\hat{f}(S) = \sum_{\ell\text{: leaf of } T} \widehat{f_\ell}(S)\,\mathrm{val}(\ell).$$

Therefore, $\hat{f}(S) = 0$ for all $S \subset [n]$ such that $|S|$ is greater than the depth of T.

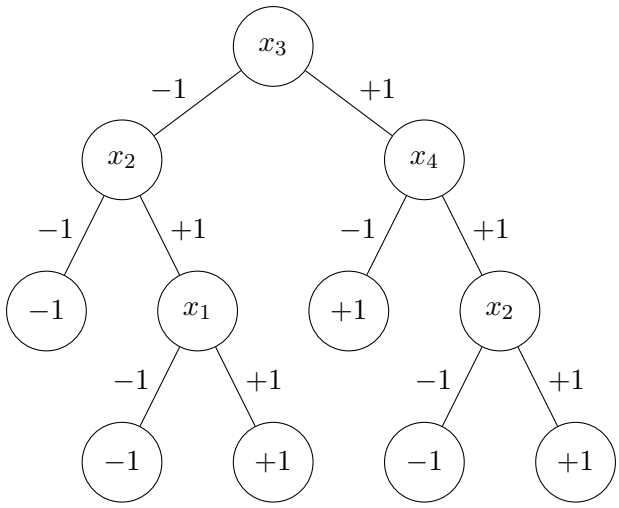

Figure 2: A decision tree.

(v) ***Constant depth circuits.*** Recall that a Boolean circuit is a DAG whose vertices are gates which represent operations (e.g., $\wedge, \vee, \neg$), constants $(1, 0)$ and variables $(x_1, \ldots, x_n)$. We allow each operation to have an unbounded number of inputs; the $\neg$ gate can have only one input. Can we compute the parity (XOR) of n bits in a constant depth with a polynomial-size input for each operation? The answe is "no," which follows from the switching lemma by Furst, Saxe and Sipser.

Theorem 7 (Hastad, Linial, Mansour and Nisan)**.** *For any function f computable via a size-s depth-d circuit,*

$$\sum_{\substack{S \subset [n] \\ |S| > t}} \hat{f}(S)^2 \leq \alpha,$$

where $t = O(\log(s/\alpha))^{d-1}$.

Taking $s = \mathrm{poly}(n)$, $d = O(1)$ and $\alpha = O(\varepsilon)$ implies $t = O(\log^d(n/\varepsilon))$. Therefore, the low degree algorithm gives an $n^{O(\log^d(n/\varepsilon))}$ sample algorithm for learning.

(vi) ***Learning half-spaces.***

Definition 8. For $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$, the function $h : \{\pm 1\}^n \to \{\pm 1\}$ defined by $h(x) = \text{sign}(w \cdot x - \theta)$ is called a *half-space function*.

Theorem 9. *A half-space function $h : \{\pm 1\}^n \to \{\pm 1\}$ has Fourier concentration $\alpha(\varepsilon) = c/\varepsilon^2$ for some constant c.*

Corollary 10. *The low degree algorithm learns half-spaces with $n^{O(1/\varepsilon^2)}$ samples.*

4.3 Noise Sensitivity

Definition 11. For $\varepsilon \in (0, 1/2)$, the *noise operator* $N_\varepsilon(x)$ randomly flips each bit of x with probability ε, given $x \in \{\pm 1\}^n$.

Definition 12. The *noise sensitivity* of a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$ is defined to be

$$\text{NS}_\varepsilon(f) = \mathop{\mathbb{P}}_{\substack{x \in \{\pm 1\}^n \\ \text{noise}}} \left[f(x) \neq f\left(N_\varepsilon(x)\right) \right].$$

Following are examples of the noise sensitivity of a Boolean function.

(i) If $f(x) = x_1$, then $\text{NS}_\varepsilon(f) = \varepsilon$.

(ii) If $f(x) = x_1 \cdots x_k$, then

$$\begin{aligned} \text{NS}_\varepsilon(f) &= \mathbb{P}[f(x) = \mathsf{F}, f(N_\varepsilon(x) = \mathsf{T})] + \mathbb{P}[f(x) = \mathsf{T}, f(N_\varepsilon(x) = \mathsf{F})] \\ &= 2\,\mathbb{P}[f(x) = \mathsf{T}, f(N_\varepsilon(x) = \mathsf{F})] \\ &= 2 \cdot \frac{1}{2^k} \cdot (1 - (1 - \varepsilon)^k). \end{aligned}$$

Therefore, if $\varepsilon \ll 1/k$, then $\text{NS}_\varepsilon(f) \approx \varepsilon k / 2^{k-1}$; if $\varepsilon \gg 1/k$, then $\text{NS}_\varepsilon(f) \approx (1 - e^{-\varepsilon k})/2^{k-1}$.

(iii) If $f(x) = \text{Maj}(x_1, \ldots, x_n)$, then $\text{NS}_\varepsilon(f) = O(\sqrt{\varepsilon})$.

To see this, note that $\text{Maj}(x)$ corresponds to a random walk on a line starting at 0, and that the location corresponds to the sum of the x_i's so far.

Fact 13. *If $X_1, \ldots, X_n \in \{\pm 1\}$ are i.i.d random variables, then $\mathbb{E}[|X_1 + \ldots + X_n|] = \sqrt{n}$, and (informally) $|X_1 + \ldots + X_n|$ is likely to be close to $\sqrt{n}$.*

Therefore, $N_\varepsilon(x)$ corresponds to a random walk on εn bits, where each flip displaces by ± 2. By Fact 13, the expected displacement is $2\sqrt{\varepsilon n}$.

Given $x \in \{\pm 1\}^n$, we consider the following process:

1. Talk a walk specified by x.
2. Continue the walk according to $2N_\varepsilon(x)$.

Pretend that the first walk leaves us at $\sqrt{n}$. By Markov's inequality,

$$\begin{aligned} \mathbb{P}[\text{the second walk takes us accross 0}] &= \frac{1}{2}\,\mathbb{P}[\text{the second displacement is greater than } \sqrt{n}] \\ &= \frac{1}{2} \cdot \frac{\mathbb{E}[\text{the second displacement}]}{\sqrt{n}} \\ &= \frac{1}{2} \cdot \frac{2\sqrt{\varepsilon n}}{\sqrt{n}} = \sqrt{\varepsilon}. \end{aligned}$$

(iv) ***Linear threshold functions (half-spaces).***

Theorem 14. *If f is a linear threshold function (i.e., a half-space), then* $\mathrm{NS}_\varepsilon(f) < 8.8\sqrt{\varepsilon}$.

(v) ***Parity functions.***

Proposition 15. *Let $S \subset [n]$ be such that $|S| = k$. Then*

$$\mathrm{NS}_\varepsilon(\chi_S) = \frac{1-(1-2\varepsilon)^k}{2}.$$