6.842 Randomness and Computation April 6, 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.