Lecture 5: The Computational Complexity of Learning

CSE 427: Machine Learning

Md Zahidul Hasan

Lecturer, Computer Science
BRAC University

Spring 2023

Efficient PAC Learning

Definition

We define a **learning problem** as a triplet $(Z, \mathcal{H}, \ell)$ where Z is a domain, $\mathcal{H}$ is a hypothesis set, and ℓ is a loss function. $\mathcal{A}$ is a learning algorithm that learns from a sample S of size m. Let's say, Z has n features. Then $\mathcal{A}$ is said to be an **efficient PAC learning algorithm** if:

1. for any $1 > \epsilon, \delta > 0$ and any distribution $\mathcal{D}$ over Z, $Pr[R(h_S) \leq \min_{h \in \mathcal{H}} R(h) + \epsilon] \geq 1 - \delta$.
2. $\mathcal{A}$ runs in $poly(\frac{1}{\epsilon}, \frac{1}{\delta}, n, |\mathcal{H}|)$.

Example

Let's say we are talking about n dimensional axis-aligned rectangles.

$$h_{a_1,a_2,\cdots,a_n,b_1,b_2,\cdots,b_n}(x_1, x_2, \cdots, x_n) = \begin{cases} 1 & \forall i \in [n], a_i \leq x_i \leq b_i \\ 0 & \text{otherwise} \end{cases}$$

Axis-aligned Rectangles: Realizable Case

Realizable Case: For every axis i, let's choose the smallest value of x_i across all m data points. Let's call it p_i and let's choose the largest value of x_i in the same dimension. Let's call it q_i. Now, if we define the rectangle as $\{(x_1, x_2, \cdots, x_n) : \forall i \in [n], x_i \in \mathbb{R} \wedge p_i \leq x_i \leq q_i\}$, then the empirical error is zero. It takes $\mathcal{O}(m)$ complexity to find the maximum and the minimum in a single dimension. The input is n dimensional. Hence the complexity of the approach is $\mathcal{O}(mn)$. Since, we know that for the problem to be PAC-learnable, $m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}$, therefore, the complexity is $\mathcal{O}(\frac{4n}{\epsilon} \log \frac{4}{\delta})$ which is a polynomial function of $\frac{1}{\epsilon}, \frac{1}{\delta}, n$. So, the problem of learning n-dimensional axis-aligned rectangles is efficiently PAC-learnable in the realizable case.

Axis-aligned Rectangles: Agnostic Case

Agnostic Case: Suppose we are given a sample of m points. Since there is no such axis-aligned rectangle that can give us 0 error in the training sample, we need to find one that minimizes the empirical error. Let's say we pick a rectangle that has no points from the sample on its boundaries. Then we can shrink this rectangle until it contains at least one point on each of its $2n$ boundaries if the rectangle contains at least one point inside its boundaries. Both rectangles have the same empirical error. Hence we can use $2n$ points to define a rectangle. So, there will be $k = \binom{m}{2n} < m^{2n}$ different rectangles. By exhaustively using each of these rectangles as a hypothesis, we can find the rectangle that minimizes the empirical error. For each of these hypotheses, it takes $\mathcal{O}(m)$ operations to check whether the m points lie inside or not. So, the overall complexity is $\binom{m}{2n} \times m$, roughly $\mathcal{O}(m^n)$. So, even though it's polynomial in m given that n is bounded. But it's still not polynomial in n. This problem is in NP.

Other Examples

Boolean Conjunctions: In this problem we are trying to learn the following boolean conjunction where for $1 \leq i \leq n$, $x_i \in \{0, 1\}$: $f(x_1, x_2, \cdots, x_n) = \phi_1(x_1) \wedge \phi_2(x_2) \wedge \cdots \wedge \phi_n(x_n)$ where each of these $\phi_i(x_i)$ either returns x_i as it is, or it negates and returns $\bar{x}_i$ or it just makes x_i disappear from this expression (or you could say it always returns 1). We are given the output of these functions for m samples. We need to learn for each i, which $\phi_i(x_i)$ does what.

- **Realizable Case:** DIY. Prove that it can be done in $\mathcal{O}(mn)$ complexity. And then express m in terms of n, δ, ϵ.
- **Agnostic Case:** No polynomial time solution unless $P = NP$.

k-Term DNF: Here, we need to learn the boolean disjunction of k terms where each of the terms are boolean conjunctions of at most n variables. For example, if $n = 3$ and $k = 2$, one such expression could be $(\bar{x}_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2 \wedge x_3)$. It has no polynomial time solution unless $NP = RP$ which means Randomized Polynomial-time.

k-CNF

k-CNF is the conjunction of an arbitrary number of terms where each term is a disjunction of at most k boolean variables. For example, $T_1 \wedge T_2 \wedge \cdots \wedge T_r$ where each T_i can have at most k attributes or their negations. T_i could be $(x_1 \vee \bar{x}_1 \vee x_4 \vee \cdots \vee \bar{x}_{k-1})$ for example. We assume that all of these T_is are unique because $a \wedge a = a$. So, even if they were not unique, they could be made unique. So, what is the maximum value of r? That's the maximum number of different boolean disjunctions possible from k variables which is $r \leq s = \sum_{i=1}^{k} \binom{2n}{i}$. Because there $2n$ boolean items namely $\{x_1, x_2, \cdots, x_n, \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n\}$. For each of the T_i, we could choose at most k of them. Let's examine the asymptotic behavior of $\sum_{i=1}^{k} \binom{N}{i}$.

$\frac{\sum_{i=1}^{k} \binom{N}{i}}{\binom{N}{K}} = 1 + \frac{k}{N-k+1} + \frac{k(k-1)}{(N-k+1)(N-k+2)} + \cdots$

$\leq 1 + \frac{k}{N-k+1} + \frac{k^2}{(N-k+1)^2} + \cdots = \frac{N-k+1}{N-2k+1}$. When N is sufficiently large, the right-hand side is 1. So, $\sum_{i=1}^{k} \binom{N}{i} = \mathcal{O}(\binom{N}{k})$.

K-CNF Continued

$\binom{N}{k}$ is exponential in N because,
$\binom{N}{k} = \frac{N-k+1}{1}\frac{N-k+2}{2}\ldots\frac{N-k+k}{k}$.
For $1 \leq x \leq k$, we have $\frac{N-k+x}{x} = \frac{N-k}{x} + 1 \leq N - k + 1 \leq N$. So, $\binom{N}{k} \leq N^k$. Therefore, $|\mathcal{H}| = 2^s = 2^{\sum_{i=1}^{k}\binom{2n}{i}} \leq 2^{(2n)^k}$.

Algorithm: Let's create a new dataset where the features are the s disjunctions of length at most k. So, if the label is 1 and if a feature has value 1, then the disjunction associated with that feature is present in our expression. Pretty easy.

Complexity: The overall complexity is $\mathcal{O}(ms) = \mathcal{O}(m(2n)^k)$ where $m = \frac{1}{\epsilon}(\log|\mathcal{H}| + \log\frac{1}{\delta}) = \frac{1}{\epsilon}(2^k n^k \log 2 + \log\frac{1}{\delta})$.

So, the final complexity is $\mathcal{O}(\frac{(2n)^k}{\epsilon}(2^k n^k \log 2 + \log\frac{1}{\delta}))$ which is a polynomial function of $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and n.

Representation Matters: K-term DNF Revisited

A K-term DNF is the disjunction of exactly k terms where each term is a conjunction of at most n boolean literals. Let's say our boolean features are $x_1, x_2, \cdots, x_n$. So, a k-term DNF would be $T_1 \vee T_2 \vee \cdots \vee T_k$ where T_k can be $x_1 \wedge x_2 \wedge \bar{x}_2 \wedge \cdots \wedge x_4$. Using associativity of $\vee$, we can express the DNF as follows: $T_1 \vee T_2 \vee \cdots \vee T_k = \bigwedge_{x_1 \in T_1, x_2 \in T_2, \cdots, x_k \in T_k} (x_1 \vee x_2 \vee \cdots \vee x_k)$. This means that any k-term DNF can be expressed as a k-CNF of r terms with $r \leq s = (2n)^k$. We know that a k-CNF can be learned with $|\mathcal{H}| = 2^{(2n)^k}$. So, the final complexity is $\mathcal{O}(\frac{(2n)^k}{\epsilon}(2^k n^k \log 2 + \log \frac{1}{\delta}))$. So, k-term DNFs can be efficiently learned which is contradictory to our previous findings. So, what's wrong? Notice that all k-term DNFs can be represented as k-CNFs but not the other way around. So, we have just used a larger hypothesis class than our initial choice. k-CNF has more representational power than k-term DNFs and has a structure that allows for efficient PAC-learning.