CSCI699: Theory of Machine Learning Fall 2021

Lecture 11: Learning Under Random Noise & Statistical Query Learning

Instructor: Vatsal Sharan *Scribe: Fatih Erdem Kizilkaya*

Recap:

The last time, we saw that

- Under some cryptographic assumptions (such as the existence of a trapdoor one way function) there exist learning problems that are hard even improperly.
- Agnostically learning even the class of halfspaces is hard over some complexity assumptions (hardness of refuting random k-XOR).
- However, halfspaces are efficiently agnostically learnable under the uniform distribution $\{\pm 1\}^d$ or the unit sphere, or the Gaussian distribution over $\mathbb{R}^d$.
- Agnostic learning allows "worst-case noise", which makes it quite demanding. For example, see the following theorem:

  Theorem 1. *There is no efficient proper learning for agnostically learning rectangles in* $\mathbb{R}^d$ *and conjunctions, unless* $\mathsf{RP} = \mathsf{P}$.

Today: We discuss two closely related models that are more resilient to "worst-case noise". Then, we show that conjunctions are efficiently learnable for these models.

1 Learning with Random Classification Noise (RCN)

Given a set of examples $\mathcal{X}$ with corresponding binary labels $\mathcal{Y}$, suppose that each label is flipped with probability η (white-noise model), i.e., only the labels are noisy. We can model this setting using the following oracle:

Example Oracle: Given a hypothesis $c(x) : \mathcal{X} \to \mathcal{Y}$ and a distribution D over $\mathcal{X}$, the oracle $\mathsf{EX}^{\eta}(c, D)$ is defined as follows:

- Draw an example $x \sim D$ from $\mathcal{X}$
- With probability $1 - \eta$, return $(x, c(x))$
- With probability η, return $(x, 1 - c(x))$

Criterion for Success: With probability $1 - \delta$, we want to find a hypothesis h such that:

$$error(h; c, D) = \Pr_{x \sim D}[c(x) \neq h(x)] \leq \varepsilon$$

where c is the hypothesis that returns the true labels.

We also restrict η to lie in $[0, 1/2)$ since learning is impossible when $\eta = 1/2$. As η approaches $1/2$ learning becomes harder, so we will give more time to the learning algorithm by allowing it to run in factors of $1/(1-2\eta)$.

Definition 2 (PAC Learning with RCN). *Let $\mathcal{C}$ be a concept class and $\mathcal{H}$ be a hypothesis class over $\mathcal{X}^d$. We say that $\mathcal{C}$ is efficiently PAC learnable with RCN if there exists a learning algorithm A such that for all $c \in \mathcal{C}$, and for all distributions D over $\mathcal{X}^d$, and for all $\varepsilon, \delta \in (0, 1/2)$ if A is given access to $\mathsf{EX}^\eta(c, D)$ and inputs ε, δ and $\eta_0 \in (\eta, 1/2)$, then A outputs h such that $error(h; c, D) \le \varepsilon$ with probability $1 - \delta$ (probability includes randomness in noisy labels).*

We say that C is efficiently PAC learnable with RCN if A also runcs in time $\mathrm{poly}(d, 1/\varepsilon, 1/\delta, 1/(1-2\eta_0))$.

Algorithm for Learning Conjunctions

The algorithm we saw for learning conjunctions in Lecture 8 fails miserably under random noise.

We now give a robust algorithm that relies on aggregate statistics instead of a single example.

Suppose that l is a literal ($\bar{x}_i$ or x_i). We define

- $P_0(l) = \mathrm{Pr}_{a\sim D}[l$ is 0 in assignment $a]$,
- $P_{bad}(l) = \mathrm{Pr}_{a\sim D}[l$ is 0 in assignment a and the true label of a is $c(a) = 1]$.

Note that if l is in conjunction $c(x)$, then $P_{bad}(l) = 0$. We say that

- l is *significant* if $P_0(l) \ge \varepsilon/8d$,
- l is *harmful* if $P_{bad}(l) \ge \varepsilon/8d$.

where d is the number of variables.

Claim: If h is conjunction of all significant literals that are not harmful, then $error(h; c, D) \le \varepsilon/2$.

Proof. We can decompose $error(h; c, D)$ into (i) and (ii) as follows:

$$error(h; c, D) = \underbrace{\mathrm{Pr}_{a\sim D}[h(a) = 1 \wedge c(a) = 0]}_{\text{(i)}} + \underbrace{\mathrm{Pr}_{a\sim D}[h(a) = 0 \wedge c(a) = 1]}_{\text{(ii)}}$$

(i) This probability corresponds to the events where $c(x)$ has a literal l that $h(x)$ does not have (and l is set to 0 in assignment a). Note that l cannot be a harmful literal by definition. Moreover, since h is conjunction of all significant literals that are not harmful, l must be insignificant. Then, since we have $2d$ literals (including negations), we get (i) $\le 2d \cdot \varepsilon/8d = \varepsilon/4$.

(ii) This probability corresponds to the events where $h(x)$ has a literal l that $c(x)$ does not have (and l is set to 0 in assignment a). By definition, l is a significant literal that is not harmful. So similarly, we get (ii) $\le 2d \cdot \varepsilon/8d = \varepsilon/4$.

Thus, $error(h; c, D) =$ (i) + (ii) $\le \varepsilon/2$. □

Note that we are still in the noiseless case. That's why we bound $error(h; c, d)$ by $\varepsilon/2$ instead of ε. We need the remaining gap for estimating $P_0(l)$ and $P_{bad}(l)$ for each literal l based on samples from $\mathsf{EX}^\eta(c, D)$, which can be done by taking a large set and get concentration with Chernoff's bound (left as an exercise).

2 Statistical Query Learning

Suppose that we do not have access to the example oracle. Instead, consider the following oracle.

Statistical Query Oracle: Given a hypothesis $c(x) : \mathcal{X} \to \mathcal{Y}$ and a distribution D over $\mathcal{X}$, the oracle $\mathsf{STAT}(c, D)$ is defined as follows:

- A query to $\mathsf{STAT}(c, D)$ is a pair (ϕ, τ) where $\phi : \mathcal{X} \times \{0, 1\} \to \{0, 1\}$ and $\tau \in (0, 1)$.
- Let $P_\phi = \Pr_{x \sim D}[\phi(x, c(x)) = 1]$
- $\mathsf{STAT}(c, D)$ returns $\hat{P}_\phi$ satisfying $\hat{P}_\phi \in [P_\phi - \tau, P_\phi + \tau]$.

Examples: Consider a spam classification task. The following might be a statistical query that we ask to the oracle:

"What fraction of e-mails labelled as spam (label is 1) have the words `Urgent` and `Free` but not the word `USC`, and the numbers of words is less than or equal to 20?"

We can also estimate $P_0(l)$ from our conjunction algorithm by query ϕ_l where

$$\phi_l(x, y) = \begin{cases} 1 & \text{if } l \text{ is } 0 \text{ in } a \\ 0 & \text{otherwise} \end{cases}$$

since $P_{\phi_l} = \mathbb{E}_{x \sim D}[\phi_l(x, c(x))] = P_0(l)$.

Similarly, we can estimate $P_{bad}(l)$ from our conjunction algorithm by query ϕ'_l where

$$\phi'_l(x, y) = \begin{cases} 1 & \text{if } l \text{ is } 0 \text{ in } a \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}$$

since $P_{\phi'_l} = \mathbb{E}_{x \sim D}[\phi'_l(x, c(x))] = P_{bad}(l)$.

Definition 3 (SQ Learning). *Let $\mathcal{C}$ be a concept class and $\mathcal{H}$ be a hypothesis class. We say that $\mathcal{C}$ is efficiently PAC learnable from statistical queries (SQ) using $\mathcal{H}$, if there exists an algorithm A, and polynomials $p(\cdot, \cdot)$, $q(\cdot, \cdot)$ and $r(\cdot, \cdot)$ such that for all $c \in \mathcal{C}$, and for all distributions D over $\mathcal{X}^d$, and for all $\varepsilon \in (0, 1/2)$, if algorithm A is given access to $\mathsf{STAT}(c, D)$ and input ε then*

- *For each query (ϕ, θ) made by algorithm A, the predicate ϕ can be evaluated in time $q(1/\varepsilon)$ and $1/\tau$ is bounded by $r(1/\varepsilon, d)$.*
- *Algorithm A runs in time $p(1/\varepsilon, d)$.*
- *Algorithm A outputs hypothesis $h \in \mathcal{H}$ such that $error(h; c, D) \leq \varepsilon$.*

Remark. *Notice that there is no confidence parameter δ in definition of SQ learning. We used δ before to take care of the probability of seeing a set of "bad examples". However, the example oracle $\mathsf{EX}(c, D)$ has been replaced by statistical query oracle $\mathsf{STAT}(c, D)$, which is deterministic.*

Theorem 4. *If a concept class $\mathcal{C}$ is efficiently SQ learnable using a hypothesis class $\mathcal{H}$, then $\mathcal{C}$ is efficiently PAC learnable using $\mathcal{H}$.*

Proof. Left as an exercise. (Hint: Implement $\mathsf{STAT}(c, D)$ using $\mathsf{EX}(c, D)$.) $\square$

Remark. *If a concept class $\mathcal{C}$ is efficiently SQ learnable, then $\mathcal{C}$ is efficiently PAC learnable with differential privacy.*

Theorem 5. *If a concept class $\mathcal{C}$ is efficiently SQ learnable, then $\mathcal{C}$ is efficiently PAC learnable with RCN.*

Theorem 6. *The class of conjunctions is efficiently learnable in the SQ model. Therefore, it is efficiently learnable with RCN.*

Proof. Our conjunction learning algorithm can be implemented in the SQ model. $\square$