COMP 585 **Spring 2024**

Probabilistic Toolkit for Learning and Computing *Instructor: Maryam Aliakbarpour*

March 28, 2024 *Scriber: Yu-Ching Shen*

Lecture X

1 PAC in statistical query model

In the statistical query model, a PAC learning algorithm can access an oracle that returns the expectation value of some functions. But the oracle is under some "noise". It does not answer the true expectation value but a value up to an additive error. The goal of the algorithm is to learn the target concept by using the oracle. We formally define the problem as follows.

Definition 1.1 (PAC learning in statistical query model)**.** *Let $\mathcal{X}$ be the instance space, $\mathcal{C}$ be the concept class, $c^* : \mathcal{X} \to \{-1, 1\} \in \mathcal{C}$ be the target concept, and $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{-1, 1\}$ satisfying that $y = c^*(x)$ if $(x, y) \sim \mathcal{D}$.*

Let $P_X := \mathbf{E}_{(x,y)\sim\mathcal{D}}[X(x, y)]$ associated with the function $X : \mathcal{X} \times \{-1, 1\} \to \{-1, 1\}$. In the statistical query model, there is an oracle O first receives a tolerant parameter $\tau \in \mathbb{R}$, and a function X, then O answers $\hat{P}_X$ such that $|\hat{P}_X - P_X| \le \tau$ for all X.

An algorithm $\mathcal{A}$ queries the oracle O on τ and $X_1, X_2, \ldots, X_t$, and then receives $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_t$. The algorithm $\mathcal{A}$ outputs a concept $\hat{c}$ such that $err(\hat{c}) \le \varepsilon$.

(The function $err(\cdot)$ is defined by $err(c) := \mathbf{Pr}_{(x,y)\sim\mathcal{D}}[c(x) \ne y]$.)

We list some remarks on the query model below.

Remark 1. *In the query model, the algorithm $\mathcal{A}$ does not draw samples from $\mathcal{D}$.*

Remark 2. *In the query model, we require the algorithm to output a concept with a small error with probability one. This is because the answer from the oracle has an error smaller than the tolerant parameter with certainty.*

Remark 3. *There are two kinds of algorithms in the query model. The first is non-adaptive algorithms, which decide the queries $X_1, X_2, \ldots, X_t$ in advance. The second is adaptive algorithms, which can choose X_i depending on $\hat{P}_{X_1}, \hat{P}_{X_2}, \ldots, \hat{P}_{X_{i-1}}$. Our analysis later works for both cases.*

2 PAC in the presence of noise

Now, we consider the PAC learning in the standard model with noise. An algorithm samples instances (x, y)'s from a "noisy" distribution $\mathcal{D}'$, where $c^*(x)$ is flipped with probability $\eta \in (0, 1/2)$. We define the problem formally as follows.

Definition 2.1. *Let $\mathcal{X}$, $\mathcal{C}$, c^*, and $\mathcal{D}$ be the same in Definition 1.1. Let $\eta \in (0, 1/2)$ and $\mathcal{D}'$ be a noisy distribution over $\mathcal{X} \times \{-1, 1\}$, where the marginal distribution over $\mathcal{X}$ is same as $\mathcal{D}$, and $y = c^*(x)$ with probability $1 - \eta$ and $y = -c^*(x)$ with probability η. An algorithm $\mathcal{B}$ draws samples from $\mathcal{D}'$ and tries to outputs a concept $\hat{c}$ such that $err(\hat{c}) \le \varepsilon$ with probability $1 - \delta$.*

We are going to show that PAC learning in the statistical query model implies PAC learning in the standard model with noise in the next section.

3 Statistical query model ⇒ standard model with noise

Assume that the algorithm in the oracle model $\mathcal{A}$ solves PAC learning problem with the parameter τ, and the functions $X_1, X_2, \ldots, X_t$. If the algorithm in the standard model $\mathcal{B}$ can compute the corresponding value $\hat{P}_{X_1}, \hat{P}_{X_2}, \ldots \hat{P}_{X_t}$, then $\mathcal{B}$ can solve the problem by running $\mathcal{A}$ with $\hat{P}_{X_1}, \hat{P}_{X_2}, \ldots \hat{P}_{X_t}$. Next, we are going to show that $\hat{P}_X$ can be computed by sampling instances from $\mathcal{D}'$. In other words, we are going to show that we can estimate $\mathbf{E}_{(x,y)\sim\mathcal{D}}[X(x, y)]$ up to an additive error τ by sampling $(x, y) \sim \mathcal{D}'$.

We have that

$$\begin{aligned}\mathbf{E}[X(x,y)] &= \mathbf{E}\big[X(x,1)\cdot \mathbb{1}_{\{c^*(x)=1\}}\big] + \mathbf{E}\big[X(x,-1)\cdot \mathbb{1}_{\{c^*(x)=-1\}}\big] \\ &= \mathbf{E}\left[X(x,1)\cdot \frac{1+c^*(x)}{2}\right] + \mathbf{E}\left[X(x,-1)\cdot \frac{1-c^*(x)}{2}\right] \\ &= \frac{1}{2}\big(\mathbf{E}[X(x,1)] + \mathbf{E}[X(x,-1)]\big) + \frac{1}{2}\big(\mathbf{E}[X(x,1)\cdot c^*(x)] - \mathbf{E}[X(x,-1)\cdot c^*(x)]\big).\end{aligned} \tag{1}$$

The randomness in Equation (1) comes from $\mathcal{D}$. The first line separates the cases $y = c^*(x) = 1$ and $y = c^*(x) = -1$. And $\mathbb{1}_{\{.\}}$ denotes the indicator. The second line replaces the indicator with c^* by finding $\mathbb{1} = \pm 1$ when $c^* = 0$ or 1. The third line separates the terms dependent on $c^*(x)$ and the terms independent of $c^*(x)$.

After splitting $y = 1$ and $y = -1$, Equation (1) does not explicitly depend on y. The first term of the third line of Equation (1) is independent of $c^*(x)$, and hence can be estimated by sampling from $\mathcal{D}_{\mathcal{X}}$, where $\mathcal{D}_{\mathcal{X}}$ denote the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$. To calculate the second term, we need to know $c^*(x)$. So, our goal now is to estimate $\mathbf{E}_{x\sim\mathcal{D}_{\mathcal{X}}}[X(x, \pm 1)\cdot c^*(x)]$ by sampling from the noisy distribution. To simplify the notation, we rewrite the above goal below.

Goal 1. *Let $\varphi : \mathcal{X} \to \{-1, 1\}$ be a function independent of y. Estimate $\mathbf{E}_{x\sim\mathcal{D}_\mathcal{X}}[\varphi(x) \cdot c^*(x)]$ up to an additive error τ with probability $1 - \delta$ by sampling $(x, y) \sim \mathcal{D}'$.*

Consider the value $\mathbf{E}_{(x,y)\sim\mathcal{D}'}[\varphi(x) \cdot y]$. We have

$$\begin{aligned}\mathbf{E}[\varphi(x) \cdot y] =& 1 \cdot \mathbf{Pr}[\varphi(x) = 1 \text{ and } y = 1] + (-1) \cdot \mathbf{Pr}[\varphi(x) = 1 \text{ and } y = -1] \\ &+ (-1) \cdot \mathbf{Pr}[\varphi(x) = -1 \text{ and } y = 1] + 1 \cdot \mathbf{Pr}[\varphi(x) = -1 \text{ and } y = -1] \\ =&\mathbf{Pr}[\varphi(x) = y] - \mathbf{Pr}[\varphi(x) \neq y]. \end{aligned} \tag{2}$$

Equation (2) holds for all distributions.

Then, we insert the value $c^*(x)$ into $\mathbf{Pr}[\varphi(x) = y]$. We have

$$\begin{aligned}\mathbf{Pr}[\varphi(x) = y] &= \mathbf{Pr}[\varphi(x) = c^*(x) \text{ and } c^*(x) = y] + \mathbf{Pr}[\varphi(x) \neq c^*(x) \text{ and } c^*(x) \neq y] \\ &= \mathbf{Pr}[\varphi(x) = c^*(x)] \cdot \mathbf{Pr}[c^*(x) = y] + \mathbf{Pr}[\varphi(x) \neq c^*(x)] \cdot \mathbf{Pr}[c^*(x) \neq y] \\ &= \mathbf{Pr}[\varphi(x) = c^*(x)](1 - \eta) + (1 - \mathbf{Pr}[\varphi(x) = c^*(x)])\eta \\ &= (1 - 2\eta)\mathbf{Pr}[\varphi(x) = c^*(x)]) + \eta. \end{aligned} \tag{3}$$

Equation (3) holds for all distributions. The second line holds because the flip occurring in the noisy distribution is independent of how we choose x such that $\varphi(x)$ equals some value.

Similarly, we have

$$\begin{aligned}\mathbf{Pr}[\varphi(x) \neq y] &= \mathbf{Pr}[\varphi(x) \neq c^*(x) \text{ and } c^*(x) = y] + \mathbf{Pr}[\varphi(x) = c^*(x) \text{ and } c^*(x) \neq y] \\ &= \mathbf{Pr}[\varphi(x) \neq c^*(x)] \cdot \mathbf{Pr}[c^*(x) = y] + \mathbf{Pr}[\varphi(x) = c^*(x)] \cdot \mathbf{Pr}[c^*(x) \neq y] \\ &= \mathbf{Pr}[\varphi(x) \neq c^*(x)](1 - \eta) + (1 - \mathbf{Pr}[\varphi(x) \neq c^*(x)])\eta \\ &= (1 - 2\eta)\mathbf{Pr}[\varphi(x) \neq c^*(x)]) + \eta. \end{aligned} \tag{4}$$

Now Equation (3) and (4) does not explicitly depend on y. Combining Equation (2), (3), and (4), we have

$$\mathbf{E}_{(x,y)\sim\mathcal{D}'}[\varphi(x) \cdot y] = (1 - 2\eta)\big(\mathbf{Pr}[\varphi(x) = c^*(x)] - \mathbf{Pr}[\varphi(x) \neq c^*(x)]\big). \tag{5}$$

By similar argument in the Equation (2), we have

$$\mathbf{E}_{x\sim\mathcal{D}_\mathcal{X}}[\varphi(x) \cdot c^*(x)] = \big(\mathbf{Pr}[\varphi(x) = c^*(x)] - \mathbf{Pr}[\varphi(x) \neq c^*(x)]\big). \tag{6}$$

Combining Equation (5) and (6), we have

$$\mathbf{E}_{x\sim\mathcal{D}_\mathcal{X}}[\varphi(x) \cdot c^*(x)] = \frac{1}{1 - 2\eta}\mathbf{E}_{(x,y)\sim\mathcal{D}'}[\varphi(x) \cdot y]. \tag{7}$$

Finally, we get the result that we want: to express $\mathbf{E}_{x\sim\mathcal{D}_\mathcal{X}}[\varphi(x) \cdot c^*(x)]$ in terms of $(x, y) \sim \mathcal{D}'$.

Now we can presents the algorithm that estimates $P_{X_1}, P_{X_2}, \dots, P_{X_t}$.

1. Inputs: $X_1, X_2, \dots, X_t$.

2. Draw $m = O(\frac{\log(t/\delta)}{\tau^2})$ samples (x_j, y_j) from $\mathcal{D}'$.
3. For $i = 1, 2, \ldots, t$, compute

$$\begin{aligned}\hat{P}_{X_i} :=& \frac{1}{m}\sum_{j=1}^{m}\frac{1}{2}\big(X_i(x_j, +1) + X_i(x_j, -1)\big) \\ &+ \frac{1}{m}\sum_{j=1}^{m}\frac{1}{2}\cdot\frac{1}{1-2\eta}\big(X_i(x_j, 1)\cdot y_j - X_i(x_j, -1)\cdot y_j\big). \end{aligned} \tag{8}$$

.

4. Output $\hat{P}_{X_1}, \hat{P}_{X_2}, \ldots \hat{P}_{X_t}$.

By Hoeffding bound, each term in Equation (8) has additive error within $O(\tau)$ with probability $1 - O(\delta/t)$, and then by union bound, $|\hat{P}_{X_i} - P_{X_i}| \leq \tau$ with probability $1 - O(\delta/t)$. We can use the same samples for each $in[t]$ because (x_j, y_j)'s are independent in each iteration. Again, use union bound over all $i \in [t]$. We have that for all $i \in [t]$, $|\hat{P}_{X_i} - P_{X_i}| \leq \tau$ with probability $1 - O(\delta)$. We conclude the discussion with the following theorem.

Theorem 4. *If there is an algorithm $\mathcal{A}$ that solves PAC learning in the statistical query model, then there is an algorithm $\mathcal{B}$ that solves PAC learning in the standard model with noise.*

4 PAC learning with unknown noise

In the previous section, we need to know the value η to estimate P_X. Does the algorithm still work when η is unknown? The answer is yes. We can guess a value of η'. As long as η' is close to η, the algorithm has a good approximation. We have

$$\left|\mathbf{E}[\varphi(x)\cdot c^*(x)] - \frac{1}{1-2\eta'}\mathbf{E}[\varphi(x)\cdot y]\right| = \left|(\frac{1}{1-2\eta} - \frac{1}{1-2\eta'})\mathbf{E}[\varphi(x)\cdot y]\right| \leq \left|(\frac{1}{1-2\eta} - \frac{1}{1-2\eta'})\right|. \tag{9}$$

The inequality holds because $\mathbf{E}[\varphi(x)\cdot y] \in [-1, +1]$. Let η_0 be the maximum of η and let $\Delta := \Theta\big(\frac{\tau}{(1-2\eta_0)^2}\big)$. If $|\eta - \eta'| \leq \Delta$, we have $\left|\big(\frac{1}{1-2\eta} - \frac{1}{1-2\eta'}\big)\right| \leq \Theta(\tau)$. (The complete derivation can be found in Section 5.4.3 in *An Introduction to Computational Learning Theory* by Michael Kearns and Umesh Vazirani.) And hence the error of empirical $\mathbf{E}[\varphi(x)\cdot c^*(x)]$ is upper bounded by $\Theta(\tau)$.

To find a η' which is Δ-close to η, we try $\eta' = 0, \Delta, 2\Delta, \ldots, \eta_0$. For each trial η', we get an output $\hat{c'}$. We can verify weather $\hat{c'}$ is that we want by checking weather $\mathbf{Pr}\Big[\hat{c'}(x) \neq y\Big]$ is Δ-close to η'.

Notice that we compute $\hat{P}_X$ by estimating $\frac{1}{1-2\eta'}\mathbf{E}[X(x, \pm 1)\cdot y]$. When η' goes larger, we need a more accurate estimation on $\mathbf{E}[X(x, \pm 1)\cdot y]$, and hence we need more samples. As a result, trying η' from 0 and increasing η' is more efficient than other ways.