CSC 665: The PAC learning model

Chicheng Zhang

September 7, 2019

1 The PAC learning model

PAC stands for "Probably Approximately Correct" [1], which is a celebrated theoretical model for studying statisical machine learning (e.g. binary classification).

Let us consider the specific setup of binary classification. Example: image classification - build a classifier that can classify images as cats or dogs.

Basic terminologies:

1. Instance domain $\mathcal{X}$ - the space where the images lie in. For example, each image is grayscale, of resolution 640×480, and is represented by its pixel intensity. Then we can take $\mathcal{X} = \mathbb{R}^{640\times 480}$.
2. Label space $\mathcal{Y} = \{-1, +1\}$ - the space where labels lie in. For example, we can use $+1$ to denote class 'cat', and -1 to denote class 'dog'.
3. Data distribution D: the distribution where examples (x, y)'s are drawn from.
4. Training set: a set S drawn iid from D.
5. Classifier h (hypothesis): a mapping from $\mathcal{X}$ to $\mathcal{Y}$ - given a image as input, the classifier outputs a label in $\{-1, +1\}$, indicating whether the image is a cat or a dog.
6. Hypothesis class $\mathcal{H}$: a (structured) collection of classifiers. For example, $\mathcal{H}$ = the set of all neural networks with ResNet-18 architecture [1], each different combination of weightings corresponds to a different classifier in $\mathcal{H}$.
7. Error: given a classifier h, define its generalization error as $\mathrm{err}(h, D) = \mathbb{P}_{(x,y)\sim D}(h(x) \neq y)$. This is the performance measure of a classifier. Define h's training error with respect to the training set S of size n as $\mathrm{err}(h, S) = \frac{1}{n}\sum_{(x,y)\in S} \mathbf{1}(h(x) \neq y)$, also abbreviated as $\mathbb{P}_{(x,y)\sim S}(h(x) \neq y)$.
8. Realizable: we are given the promise that there exists a classifier $h^\star$ in $\mathcal{H}$, such that $\mathrm{err}(h^\star, D) = 0$. Agnostic: there may or may not exists a classifier in $\mathcal{H}$ that has zero generalization error.

Definition 1. *We call an algorithm $\mathcal{A}$ PAC learns hypothesis class $\mathcal{H}$ with sample complexity function $m : (0, 1) \times (0, 1) \to \mathbb{N}$, if for any distribution D realizable with respect to $\mathcal{H}$, $\epsilon > 0$, $\delta > 0$, when $\mathcal{A}$ receives $m \geq m(\epsilon, \delta)$ iid training examples from D as input, it outputs a classifier $\hat{h}$, such that with probability $1 - \delta$,*

$$\mathrm{err}(\hat{h}, D) \leq \epsilon.$$

Remarks:

1. ϵ is called the *target error rate*, and δ is called the *confidence parameter* [2]. "Probably" means that the algorithm succeed with high probability $(1-\delta)$, and "Approximately Correct" means that the classifier returned has small error rate (ϵ).

[1] With an additional linear threshold layer for the output.

[2] This may remind you the notion of $(1 - \alpha)$-confidence interval in statistics, where the goal is to construct an interval such that it contains the underlying parameter with probability $1 - \alpha$. Here α is essentially δ, the failure probability.

2. The original definition in [1] requires that the learning algorithm runs in polynomial time. Here we relax the original definition and do not require computational efficiency.

3. The sample complexity is hypothesis-class dependent but *distribution independent*. In practice, one might want a sample complexity definition that "exploits the easiness of the data", i.e. if distribution D is easy to learn, then the learning algorithm may draw less samples from D to learn a good classifier. We will return to this point when we study model selection.

4. If $\mathcal{A}$ always return a h in $\mathcal{H}$, then it is called a *proper* learning algorithm. Otherwise, it is called an improper learning algorithm. Interestingly, improper learning can sometimes provide more power in achieving learnability and saving sample complexity.

1.1 Sample complexity of finite hypothesis classes

Let us consider the setting where the hypothesis class $\mathcal{H}$ is of finite size. In fact, it can be argued that all hypothesis classes considered in computer-based applications are finite! To see this, consider for example the set of all ResNet-18 networks, with each weight taking values in 64-bit floating-point numbers. Any such network can be represented using $(64 \times \#\text{weights})$ bits.

Here we show a fundational result: any finite hypothesis class $\mathcal{H}$ will have a PAC sample complexity of order $m(\epsilon, \delta) = \frac{1}{\epsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta})$. To see this, let us consider the following simple learning algorithm.

Algorithm 1 The consistency algorithm

Require: Training samples S of size m iid from D, hypothesis class $\mathcal{H}$

1: Find $\hat{h}$ in $\mathcal{H}$ such that $\hat{h}$ agrees with all examples in S, that is, for all (x, y) in S, $\hat{h}(x) = y$.

2: **return** $\hat{h}$.

Theorem 1. *Suppose $\mathcal{H}$ is finite. If the consistency algorithm (Algorithm 1) is given m iid examples from D realizable with respect to $\mathcal{H}$, then with probability $1 - \delta$, its output $\hat{h}$ is such that*

$$\text{err}(\hat{h}, D) \leq \epsilon_R(m, \delta, \ln |\mathcal{H}|) \triangleq \frac{\ln |\mathcal{H}| + \ln \frac{1}{\delta}}{m}. \tag{1}$$

In other words, it PAC learns hypothesis class $\mathcal{H}$ with sample complexity $m(\epsilon, \delta) = \frac{1}{\epsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta})$.

We make the following obervations on the error upper bound (Equation (1)). First, the dependence on sample size is of order $\frac{1}{m}$, which is "polynomially decreasing" in m. Second, the dependency on the size of the hypothesis class is only logarithmic. This is somewhat desirable, as $|\mathcal{H}|$ is usually exponential in the number of parameters in $\mathcal{H}$, and taking a logarithm will bring the dependency on the number of parameters from exponential to linear[3]. Third, the dependency on δ is only $\ln \frac{1}{\delta}$ - this means that the failure probability δ can be taken as a small number (such as m^{-5}) without hurting much on error bound.

Proof of Theorem 1. First, under realizable assumption, $h^\star$ has zero error with respect to D. Therefore, (with probability 1) $h^\star$ will also agree with all examples in S. This implies that Step 1 will successfully find a classifier $\hat{h}$ in $\mathcal{H}$ that agrees with all exmaples in S.

Second, let $\mu := \epsilon_R(m, \delta, |\mathcal{H}|)$. Let us consider all classifiers in $\mathcal{H}$ that has error $> \mu$ - define such subset as $\mathcal{H}_\mu$. We argue that with probability $1 - \delta$, all members of $\mathcal{H}_\mu$ has nonzero error in S.

Fix h in $\mathcal{H}_\mu$. Denote by the event E_h that h has zero error in S. Then

$$\mathbb{P}(E_h) = \prod_{i=1}^{m} \mathbb{P}(h(x_i) = y_i) = \mathbb{P}(h(x) = y)^m = (1 - \text{err}(h, D))^m < (1 - \mu)^m \leq e^{-\mu m}.$$

[3]In modern applications such as learning overparameterized neural networks, this bound is not desirable - the number of parameters can be much greater than the training sample size, which makes this bound vacuous (≥ 1).

Define $E := \cup_{h \in \mathcal{H}_\mu} E_h$. By union bound over all h in $\mathcal{H}_\mu$, we have

$$\mathbb{P}(E) \leq \sum_{h \in \mathcal{H}_\mu} \mathbb{P}(E_h) \leq |\mathcal{H}_\mu| e^{-\mu m} \leq |\,\mathcal{H}\,| e^{-\mu m}.$$

Plugging in the value of $\mu = \frac{\ln |\,\mathcal{H}\,| + \ln \frac{1}{\delta}}{m}$, we get that $\mathbb{P}(E)$ is at most $|\,\mathcal{H}\,| e^{-(\ln |\,\mathcal{H}\,| + \ln \frac{1}{\delta})} = \delta$.

Event E states that there exists h in $\mathcal{H}_\mu$ that agrees with S. Its complement $\bar{E}$ states that for all h in $\mathcal{H}_\mu$, h does not agree with S. Formally,

$$\text{for all } h \text{ in } \mathcal{H} : \text{err}(h, D) > \mu \Rightarrow \text{err}(h, S) > 0.$$

Taking the contrapositive of the statement, we have:

$$\text{for all } h \text{ in } \mathcal{H} : \text{err}(h, S) = 0 \Rightarrow \text{err}(h, D) \leq \mu. \tag{2}$$

Event $\bar{E}$ happens with probability at least $1 - \delta$.

Suppose event $\bar{E}$ happens. In this case, as the output $\hat{h}$ agrees with S, from Equation (2), $\hat{h}$ must have error at most μ. This concludes the first part of the theorem.

The sample complexity bound directly follows by observing that as long as $m \geq \frac{1}{\epsilon}(\ln |\,\mathcal{H}\,| + \ln \frac{1}{\delta})$, the $(1-\delta)$-probability error bound $\epsilon_R(m, \delta, \ln |\,\mathcal{H}\,|)$ is at most ϵ. □

2 The Agnostic PAC learning model

One clear limitation of the PAC learning model is that it assumes that the data is perfectly separable by some classifier in a given hypothesis class. Therefore, it cannot handle realistic settings where the data is noisy, or settings when no classifier in the given hypothesis class is able to fully capture the patterns in the data. The agnostic PAC model generalizes the PAC model by allowing the optimal classifer in $\mathcal{H}$ to have nonzero error.

Definition 2. *We call an algorithm $\mathcal{A}$ agnostic PAC learns hypothesis class $\mathcal{H}$ with sample complexity function $m : (0,1) \times (0,1) \to \mathbb{N}$, if for any distribution D, $\epsilon > 0$, $\delta > 0$, when $\mathcal{A}$ receives $m \geq m(\epsilon, \delta)$ iid training examples from D as input, and outputs a classifier $\hat{h}$, such that with probability $1 - \delta$,*

$$\text{err}(\hat{h}, D) \leq \min_{h' \in \mathcal{H}} \text{err}(h', D) + \epsilon.$$

The goal of Agnostic PAC model is less ambitious: it declares the success of the learner so long as it outputs a classifier whose *excess error* is at most ϵ; here the excess error of h is defined as $\text{err}(h, D) - \min_{h' \in \mathcal{H}} \text{err}(h', D)$, that is, the difference between the error of the output classifer and the optimal error in $\mathcal{H}$. On the other hand, it generalizes the PAC learning model: if the data distribution D is realizable wrt $\mathcal{H}$, the optimal error is zero, and the algorithm will return an classifier with at most ϵ error with probability $1 - \delta$.

2.1 Agnostic sample complexity of finite hypothesis classes

For the agnostic case, we consider the following generalization of the consistency algorithm, namely, empirical risk minimization. The name comes from the fact that instead of looking for a classifier that perfect fits the data, the algorithm simply finds a classifier in $\mathcal{H}$ that minimizes the training error (empirical risk).

Theorem 2. *Suppose $\mathcal{H}$ is finite. If the consistency algorithm (Algorithm 2) is given m iid examples from D realizable with respect to $\mathcal{H}$, then with probability $1 - \delta$, its output $\hat{h}$ is such that*

$$\text{err}(\hat{h}, D) \leq \min_{h \in \mathcal{H}} \text{err}(h, D) + \epsilon_A(m, \delta, \ln |\,\mathcal{H}\,|) \triangleq \sqrt{\frac{2(\ln |\,\mathcal{H}\,| + \ln \frac{2}{\delta})}{m}}. \tag{3}$$

In other words, it agnostic PAC learns hypothesis class $\mathcal{H}$ with sample complexity $m(\epsilon, \delta) = \frac{2}{\epsilon^2}(\ln |\,\mathcal{H}\,| + \ln \frac{2}{\delta})$.

Algorithm 2 Empirical risk minimzation

Require: Training samples S of size m iid from D, hypothesis class $\mathcal{H}$

1: Find $\hat{h}$ such that $\hat{h}$ has the lowest training error among $\mathcal{H}$: $\hat{h} \in \operatorname{argmin}_{h \in \mathcal{H}} \operatorname{err}(h, S)$.

2: **return** $\hat{h}$.

Proof. Let $\mu = \frac{1}{2}\epsilon_A(m, \delta, \ln |\mathcal{H}|) = \sqrt{\frac{\ln |\mathcal{H}| + \ln \frac{2}{\delta}}{2m}}$.

For all classifiers h in $\mathcal{H}$, define E_h as the event that

$$|\operatorname{err}(h, S) - \operatorname{err}(h, D)| > \mu.$$

In addition, define $E = \cup_{h \in \mathcal{H}} E_h$.

By Hoeffding's Inequality,

$$\mathbb{P}(E_h) \leq 2e^{-2m\mu^2}.$$

Therefore, by union bound and expanding the definition of μ, we get that

$$\mathbb{P}(E) \leq \sum_{h \in \mathcal{H}} \mathbb{P}(E_h) \leq |\mathcal{H}| \cdot 2e^{-2m\mu^2} \leq \delta.$$

Hence, the complement event, $\bar{E}$, happens with probability at least $1 - \delta$. For the rest of the proof, suppose event $\bar{E}$ happens.

In $\bar{E}$, we have that for all h in $\mathcal{H}$,

$$|\operatorname{err}(h, S) - \operatorname{err}(h, D)| \leq \mu. \tag{4}$$

Denote by $h^\star$ one of the classifier in $\mathcal{H}$ that has the smallest generalization error. From Equation 4 applied to the output classifier $\hat{h}$ and the optimal classifier $h^\star$, we have $|\operatorname{err}(\hat{h}, S) - \operatorname{err}(\hat{h}, D)| \leq \mu$ and $|\operatorname{err}(h^\star, S) - \operatorname{err}(h^\star, D)| \leq \mu$. In addition, as $\hat{h}$ minimizes the training error, $\operatorname{err}(\hat{h}, S) \leq \operatorname{err}(h^\star, S)$.

Therefore,

$$\begin{aligned}
\operatorname{err}(\hat{h}, D) &\leq \operatorname{err}(\hat{h}, S) + \mu && \text{Equation (4) on } \hat{h} \\
&\leq err(h^\star, S) + \mu && \text{Optimality of } \hat{h} \\
&\leq err(h^\star, D) + 2\mu && \text{Equation (4) on } h^\star \\
&\leq \min_{h \in \mathcal{H}} err(h, D) + \epsilon_A(m, \delta, \ln |\mathcal{H}|). && \text{Optimality of } h^\star \text{ and definition of } \mu
\end{aligned}$$

The sample complexity bound directly follows by observing that as long as $m \geq \frac{2}{\epsilon^2}(\ln |\mathcal{H}| + \ln \frac{2}{\delta})$, the $(1 - \delta)$-probability error bound $\epsilon_A(m, \delta, \ln |\mathcal{H}|)$ is at most ϵ. □

References

[1] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.