PAC Learnability of Finite Hypothesis Class

Shuyi Tan

December 11, 2020

1 Background

By our nature of intelligence being, we have the intuitive motivation of ··learning". The basic mechanism of statistical learning is the ability for humans and other animals to extract statistical regularities from the world around them to learn about the environment. In this project, we will explore the learnability of a hypothesis class through a foundation algorithm in the universal learning theory: Empirical Risk Minimization by walking through the problem setting and theory construction

1.1 General Learning Problem Setting

There are many classic learning problems utilized in the real life, such as classifying set of emails into spam/not spam, optimal portfolio selection, and weather forecasting. In the basic statistical setting, a learning problem is characterized by the following components:

- **Domain Set** χ: A arbitrary set χ that is the set of objects that we wish to classify or label. Mostly we concerned with the case where $\chi = \mathbb{R}^d$ or $\chi = \{0,1\}^d$, in which each object that we want to label is characterized by d real numbers. The attributes are referred to as features.
- **Label Set** $\mathcal{Y}$: The set $\mathcal{Y}$ describe the labels, which will be used for learners to assign for each $x \in \mathcal{X}$ a label $y \in \mathcal{Y}$.

We assumes that there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ that generates labeled examples. For example, pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

1.2 Elements of A statistical Learning Model

A statistical learning model usually consists of the following elements:

- **Input** A set of labeled examples in the training set $S = (x_1, y_1), ..., (x_m, y_m) \subseteq \mathcal{X} \times \mathcal{Y}$. This is the input that is known, and we denote the size of training set as m.
- **Output**: A prediction rule/hypothes is $\mathcal{H} : h : \mathcal{X} \to \mathcal{Y}$, which turns unlabeled samples to labels.
- **Data**: As we mentioned above, we assumes that exists an unknown distribution $\mathcal{D}$ over $\mathcal{X}$ and the label is generated by some mapping $f : \mathcal{X} \to \mathcal{Y}$. S is created by iid samples from $\mathcal{D}$, and they are classified by f.
- **Performance metric**: The risk/generalization error. Given a distribution D over $\mathcal{X}$, the learner is required to give a hypothesis $h : \mathcal{X} \to \mathcal{Y}$. This functions is called predictor or a classifier. The quality of each hypothesis $h \subseteq \mathcal{H}$ is measured by its $\text{err}(h) = \mathbb{P}_{x\sim D}[h(x) \neq f(x)] = \mathbb{E}_{x\sim D}[|h(x) - f(x)|]$. Ideally, we are interested in picking $h \in \mathcal{H}$ whose risk is as close as possible to $\inf_{h\in\mathcal{H}} \text{err}(h)$. Since the distribution $\mathcal{D}$ is unknown, we cannot do this directly, but we can rely on our empirical training sample $\mathcal{S}$. Generally, we expect the approximation to get better performance with larger sample size. A learning rule that allow us to choose such h are said to be consistent. [3]

1.3 The Probably Approximately Correct Learning (PAC)

We will utilize the concept of the PAC learning when exploring the learnability of hypothesis class. We start by proposing a simple example. Let our domain set $\mathcal{X}$ as a line, by placing a threshold $\theta \in \mathbb{R}$ on the line, we have

$$h_\theta(x) = \begin{cases} y_i & \text{if } x \text{ in the left hand side of } \theta \\ 0 & \text{if } x \text{ in the right hand side of } \theta \end{cases}$$

We can easily observe that the data following a uniform distribution $\mathcal{D}$, however, the information of distribution $\mathcal{D}$ is blind to our algorithm. Our goal for the algorithm is to learn the threshold r that can classify the data points into two classes with error less than a given constant ε. In this situation, there are a few facts that we need to aware. One is that, with a finite empirical training set, it is almost impossible to achieve error-free classification. Therefore, we will turn to pursue an approximately correct output. The other one would be it is possible that a correct output is only probably correct instead of being truly correct, because there exist a situation in which the training set is not representative of the whole data.

Definition 1. *(The PAC Learning) An algorithm $\mathcal{L}$ PAC learns a unknown f if $\forall \mathcal{D}$ $\exists \varepsilon > 0$ and $\exists \delta > 0$. Let $h_{\mathcal{L}}$ be the function that $\mathcal{L}$ output, then*

$$P(\text{err}(h_{\mathcal{L}}) \leq \varepsilon) \geq 1 - \delta$$

An extremely important lemma that assists with the PAC learning is about the size the training set that can achieve the PAC learning.

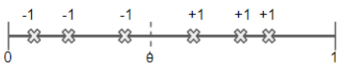
Lemma 1. *To achieve a PAC guarantee, the following condition holds true for the sample size m:*

$$m \leq \frac{1}{\varepsilon} \log(\frac{2}{\delta})$$

Proof. Pranjal Awasthi presents an detailed solution shown as follows.[1] We scan the training samples from left to right and put a threshold r anywhere in the region where there is a change in label like (A) in figure 1. Let the sample size as m, if θ lies in the region R, then we can be confident that $\text{err}(h) \leq \varepsilon$. A necessary condition is that the training set must have at least 2 examples from the region R with one with each class respectively. Since the probability of a single data point in the training set is not within $(\theta - \varepsilon, \theta)$ is less or equal to $1 - \varepsilon$, and the probability of that is not within $(\theta, \theta + \varepsilon)$ is also less or equal to $1 - \varepsilon$. Because of independence of each data point, we won't see at least one example from both regions is upper bounded by $2(1 - \varepsilon)^m$. To achieve the PAC learning, the sample size m must satisfy the following condition:

$$\begin{aligned} 2(1-\varepsilon)^m &\leq \delta \\ \rightarrow m &\leq \frac{1}{\varepsilon} \log(\frac{2}{\delta}) \end{aligned}$$

[A]



[B]

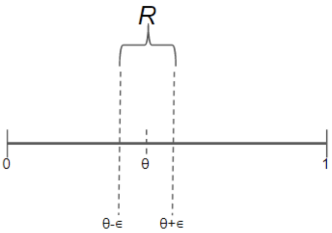


Figure 1: PAC Learning

□

1.4 The Empirical Risk Minimization

Recalling from the previous two-class example, we have the prior knowledge that the data is generated form a uniform distribution. Therefore, we roughly have an initial judge of how the threshold function will look like, which is impossible for most situations in practice. Therefore, we are interested in finding an generic algorithm that fits any types of functions. Our objective is to obtain a hypothesis that minimizes the empirical error on the training set where we have a finite number of hypotheses, and at least one of which has zero expected risk.

Definition 2. ***Empirical Risk Minimization****) The empirical risk minimization algorithm $h_{ERM} : X \to Y$ is the classifier that achieves the minimum error on the test set S. Suppose $h* \in \mathcal{H}$ is the true function that describes the data (error euqal 0), where $\mathcal{H}$ is finite. Given a training set $\mathcal{S} = \{s_1, ..., s_m\} = \{(x_1, y_i), ..., (x_m, y_m)\}$, the function returned by the ERM algorithm will be*

$$h_{ERM} = \text{argmin}_{h \in \mathcal{H}} \, \text{err}_{\mathcal{S}}(h)$$

where $\text{err}_{\mathcal{S}}(h)$ is the fraction of incorrect classification that h makes on the training set. Namely, h_{ERM} is the optimal function that produces the lowest error on the training set.

2 Theorem and Results

2.1 Finite Classes are PAC Learnable through ERM

Let $\mathcal{H}$ be a finite hypothesis class. For example, $\mathcal{H}$ can be the set of all predictors that can be implemented by a C++ program whose length is at most k bits. The learning algorithm is allowed to use the training set for determining which predictor to choose from the hypothesis class H. In particular, we will analyze the performance of the ERM learning rule:

$$h_{\mathcal{S}} \in \operatorname{argmin}_{h\in\mathcal{H}} \operatorname{err}_{\mathcal{S}}(h).$$

This restriction will raise an issue for hypothesis that does not generalize well. A potential solution is to add a additional assumption:

Definition 3. ***(Realizability assumption)*** *$\exists f \in \mathcal{H}$ such that $\operatorname{err}_{\mathcal{D}}(f) = 0$. This assumption implies that for any training set $\mathcal{S}$, we have $\operatorname{err}_{\mathcal{S}}(f) = 0$ with probability 1.[4]*

From the realizability assumption and the definition of the ERM $h_{ERM} = \operatorname{argmin}_{h\in\mathcal{H}} \operatorname{err}_{\mathcal{S}}(h)$, we have that $\operatorname{err}_{\mathcal{S}}(h) = 0$ with probability 1. We are interested in the generalization error err_D. Since err_D depends on the training set, we will analyze the probability to sample a training set for which $\operatorname{err}_{\mathcal{D}}(h)$ is within a proper range. Let ε be the accuracy parameter, and we will interpret the event $\operatorname{err}_{\mathcal{D}}(h) > \varepsilon$ as severe overfitting. If $\operatorname{err}_{\mathcal{D}}(h) \leq \varepsilon$, we regard the output of the algorithm to be approximately correct predictor as what we define as PAC in the background section. Therefore, we are interested in calculating

$$P_{\mathcal{S}\sim\mathcal{D}^m}([\operatorname{err}_{\mathcal{D}}(h) > \varepsilon])$$

Let E_B be the set of "bad" hypotheses, that is $E_B = \{h \in \mathcal{H} : \operatorname{err}_{\mathcal{D}}(h) > \varepsilon\}$. The realizability assumption implies that $\operatorname{err}_{\mathcal{S}}(h_{\mathcal{S}}) = 0$ with probability 1. We can infer that the event E_B can only happen if for some $h \in E_B$ we have $\operatorname{err}_S(h) = 0$. Therefore, the set $\mathcal{S} : \operatorname{err}_D(h_{\mathcal{S}}) > \varepsilon$ is contained in the set $\mathcal{S} : h \in \mathcal{E}_B, \operatorname{err}_S(h) = 0$. Therefore,

$$\underset{S\sim D^m}{P}\left[\operatorname{err}_D\left(h_{\mathcal{S}}\right) > \epsilon\right] \leq \underset{S\sim D^m}{P}\left[\exists h \in E_B : \operatorname{err}_S(h) = 0\right]$$

Lemma 2. ***(Union Bound)*** *Let A_1 , . . . , A_t be some events then $P(\bigcup_{i=1}^{t} A_i) \leq \Sigma_{i=1}^{t} P(A_i)$*

Rewriting $\{S : \exists h \in \mathcal{E}_B, \operatorname{err}_S(h) = 0\}$ as $\cup_{h\in\mathcal{E}_B} \{S : \operatorname{err}_S(h) = 0\}$, we apply union bound to the right-hand side of the previous inequality to get that

$$\underset{S\sim D^m}{P}\left[\operatorname{err}_D\left(h_S\right) > \epsilon\right] \leq \sum_{h\in\mathcal{H}_B} \underset{S\sim D^m}{P}\left[\operatorname{err}_S(h) = 0\right]$$

For some fixed bad hypothesis $h \in \mathcal{E}_B$, each individual element of the training set we have,

$$\underset{(\mathbf{x}_i,y_i)\sim\mathcal{D}}{P}\left[h\left(\mathbf{x}_i\right) = y_i\right] = 1 - \operatorname{err}_{\mathcal{D}}(h) \leq 1 - \epsilon$$

since the examples in the training set are sampled i.i.d. we get that for all $h \in \mathcal{E}_B$

$$\underset{S\sim D^m}{P}\left[\operatorname{err}_S(h_{\mathcal{S}}) = 0\right] = \prod_{i=1}^{m} \underset{(\mathbf{x}_i,y_i)\sim\mathcal{D}}{P}\left[h\left(\mathbf{x}_i\right) = y_i\right] \leq (1-\varepsilon)^m$$

Comparing two inequalities:

$$\begin{cases} \underset{S\sim D^m}{P}\left[\operatorname{err}_D\left(h\right) > \epsilon\right] \leq \sum_{h\in\mathcal{E}_B} \underset{S\sim D^m}{P}\left[\operatorname{err}_S(h) = 0\right] \\ \underset{S\sim D^m}{P}\left[\operatorname{err}_S(h) = 0\right] \leq (1-\varepsilon)^m \end{cases}$$

and using the inequality $1 - \varepsilon \leq e^{-\varepsilon}$ we conclude that

$$\underset{S\sim D^m}{P}\left[\operatorname{err}_D\left(h\right) > \varepsilon\right] \leq |\mathcal{E}_B|\left(1-\varepsilon\right)^m \leq |\mathcal{H}|e^{-\varepsilon m}$$

Corollary 1. *Let $\mathcal{H}$ be a finite hypothesis class, ERM PAC learns any $f \in \mathcal{H}$ provided $m \geq \frac{1}{\varepsilon}\ln(\frac{|\mathcal{H}|}{\delta})$ with $\delta in(0,1)$ and $\varepsilon > 0$.*

Proof. Let E_B be the set of "bad" hypothesis. For the h within $\mathcal{E}_B$, if $\text{err}(h) > \varepsilon$, each time we draw a random example, h has a probability greater than ε of making an error because $P_{x\sim D}(h(x) \neq f(x)) = \text{err}(h) > \varepsilon$. The probability that h looks good on any particular element of the training set fooling the algorithm is less than $1-\varepsilon$. Since the elements in the training set are chosen randomly, the probability that one bad h fools the algorithm on all the training set is that:

$$P(E_h) \leq (1-\varepsilon)^m \leq e^{-\varepsilon m}$$

If there are multiple bad functions, using the union bound, we have

$$\begin{aligned} P(\text{Multiple bad functions}) &= P(\bigcup_{h\in\mathcal{H}}) \\ &\leq \Sigma_{h\in\mathcal{H}} P(E_h) \\ &\leq |\mathcal{H}| e^{\varepsilon m} \end{aligned}$$

If we want to succeed with probability more than $1-\delta$, we can have

$$\begin{aligned} &|H|e^{-\epsilon m} < \delta \\ \Rightarrow\ &e^{-\epsilon m} < \frac{\delta}{|H|} \\ \Rightarrow\ &m \geq \frac{1}{\epsilon}\ln\frac{|H|}{\delta} \end{aligned}$$

□

By rearranging this inequality, we can quantify the uncertainty of our algorithm's in success on the training set. In other words, for a training set of fixed size m, we use the ERM algorithm to obtain $E\hat{R}M$ with $\text{err}(E\hat{R}M) = 0$. With respect to the probability, $P(\text{err}(h) \leq \frac{1}{m}\log\frac{|\mathcal{H}|}{\delta}) \geq 1-\delta$.

There are a number of limitations of ERM PAC learning finite function classes. Typically $\mathcal{H}$ is unknown, because we usually do not have prior knowledge of the class of possible function. In the meantime, it would be difficult to find the best fit $\hat{h}$. Even though ERM indicates that $\hat{h} = \text{argmin}_{h\in\mathcal{H}}\,\text{err}_{\mathcal{S}}(h)$, while we do not have clear picture of what clever algorithm to use until really trying on a couple of potential algorithms.

2.2 Learning any function

Recall that in the previous section, the PAC guarantee is based on an important assumption: $f \in \mathcal{H}$, which is formally defined as the realizability assumption in the statistical learning theory. However, the realizability assumption prevents the algorithm from handling noisy data and requires us to choose $\mathcal{H}$ very carefully. This leads us to consider: what if we remove the realizability assumption? The worst situations we should consider is that there is no $h \in \mathcal{H}$ as f, or perhaps all $h \in \mathcal{H}$ satisfy $\text{err}_{\mathcal{S}} - \text{err}(h) < \varepsilon$. The following theorem demonstrates how ERM learns:

Theorem 1. *[2] Let $\mathcal{H}$ be finite hypothesis class and $\mathcal{S}$ be the training set. Suppose $|\mathcal{S}| = m$ and $\hat{h} = \text{argmin}_{h\in\mathcal{H}}\,\text{err}_{\mathcal{S}}(h)$. Then for any $\delta \in (0,1)$, for any $\varepsilon > 0$,*

1. *if $m \geq \frac{2}{\varepsilon^2}log(\frac{2|\mathcal{H}|}{\delta})$, then for any h, $P(\text{err}_{\mathcal{S}}(h) - \text{err}(h) \leq \varepsilon) \geq 1-\delta$.*
2. *$P(\text{err}(\hat{h}) \leq \text{h}_{\text{ERM}}) + 2\varepsilon) \geq 1-\delta$ where $h_{ERM} = \text{argmin}_{h\in\mathcal{H}}\,\text{h}$.*

The result follows from a well-known principle: Hoeffding's inequality.

Lemma 3. *(Hoeffding's Inequality)[5] Let $X_1, ..., X_m$ be iid random variables such that $a_i < X < b_i$ with probability 1. Let $S_n = \Sigma_{i=1}^n X_i$, then for any $t > 0$,*

$$P(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\Sigma_{i=1}^n (b_i - a_i^2)}}$$

The intuition for the Hoeffding's Inequality is very simple. We have a bunch of variable X_i, and we know that when we average a bunch of them up, what we usually get the something close to expected value. The proof of Hoeffding's inequality will be in the appendix. Rewriting the Hoeffding's Inequality in the following form:

$$\mathbb{P}(|\bar{X} - \mathbb{E}(X)| \geq t) \leq 2e^{-2nt^2}$$

We will prove two parts of theorem 2 respectively.

Proof. (Theorem 1.1) We would like to bound the probability that some function $h \in \mathcal{H}$ is a 'bad" function. Recall that a 'bad' function $h \in \mathcal{H}$ is function with $\text{err}_{\mathcal{S}}(h) - \text{err}(h) > \varepsilon$. Namely, bad performance on the whole set/test set. We want to prove the opposite version of $\forall h, P(\text{err}_{\mathcal{S}}(h) - \text{err}(h) \leq \varepsilon) \geq 1 - \delta$: $\exists h$, $P(\text{err}_{\mathcal{S}}(h) - \text{err}(h) \leq \varepsilon) \leq \delta$. Starting from a fixed function $h \in \mathcal{H}$,

$$\text{P}\left[|\text{err}_S(h) - \text{err}(h)| > \epsilon\right] = \text{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_i - \text{err}(h)\right| > \epsilon\right]$$

where m is the size of the training set $\mathcal{S}$,and $\mathbf{1}_i$ is an indicator variable where

$$\mathbf{1}_i = \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{otherwise} \end{cases}$$

We know that the expected value of $\mathbf{1}_i$ is $\text{err}(h)$, so $E(\frac{1}{m}\Sigma_{i=1}^n \mathbf{1}_i) = \text{err}(h)$. Using Hoeffding's inequality, we can obtain

$$\text{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_i - \text{err}(h)\right| > \epsilon\right] \leq 2e^{-2m\epsilon^2}$$

Using the union bound, we have: there exist h such that

$$\text{P}\left[\exists h \in H : |\text{err}_S(h) - \text{err}(h)| > \epsilon\right] \leq |H| \left(2e^{-2m\epsilon^2}\right) \leq \delta$$

Rewriting this inequality, we can get

$$m \leq \frac{1}{2\varepsilon^2} log(\frac{2|\mathcal{H}|}{\delta})$$

□

Now let's prove the second part of theorem 1.

Proof. ((Theorem 1.2)) Let $\hat{h}$ is the function returned by ERM, namely $\hat{h} = \text{argmin}_{h \in \mathcal{H}} \text{err}_{\mathcal{S}}(h)$. From the first part of the proof, we have $\text{err}(\hat{h}) \leq \text{err}_{\mathcal{S}}(\hat{h}) + \varepsilon$. By the definition of $\hat{h}$, $\text{err}(\hat{h}) \leq \text{err}_{\mathcal{S}}(h_{ERM}) + \varepsilon$. Then apply theorem 2.1 to h_{ERM}, we can acquire $\hat{\text{h}} \leq \text{err}(h_{ERM}) + 2\varepsilon$ □

Developed from theorem 2, we have an additional corollary for finite class.

Corollary 2. *Given a finite function class $\mathcal{H}$, for any $\varepsilon > 0$, $\delta > 0$, any distribution $\mathcal{D}$ over $\mathcal{X}$ and any target function $h*$, let m be the size the training set $\mathcal{S}$, then for any $h \in \mathcal{H}$,*

$$\forall h \in H \, |\text{err}_S(h) - \text{err}(h)| \leq \sqrt{\frac{1}{2m} \log\left(\frac{2|H|}{\delta}\right)}$$

Proof. (Corollary 2) Recall that there exist $h \in \mathcal{H}$ such that $P(|\operatorname{err}_{\mathcal{S}}(h) - \operatorname{err}(h)| > \varepsilon)$. Applying Hoeffding's inequality to $\varepsilon = \sqrt{\frac{1}{2m} log(\frac{2|\mathcal{H}|}{\delta})}$, and converting $1 - P(h \in \mathcal{H} | \operatorname{err}_{\mathcal{S}}(h) - \operatorname{err}(h)| > \varepsilon)$ to $P(\forall h \in \mathcal{H} | \operatorname{err}_{\mathcal{S}}(h) - \operatorname{err}(h)| > \varepsilon)$, we can obtain:

$$\forall h \in H \, |\operatorname{err}_S(h) - \operatorname{err}(h)| \leq \sqrt{\frac{1}{2m} \log\left(\frac{2|H|}{\delta}\right)}$$

□

3 Open questions and research directions

3.1 Is small or big hypothesis class better?

In corollary 1, we propose that if we are able to find a hypothesis $h \in \mathcal{H}$ where H has finite with mw independent random labeled training examples, then for any strictly positive pair (δ, ε) we can assert with probability $1 - \delta$ that the error of h is less than provided that:

$$m \geq \frac{\ln(|\mathcal{H}|) + \ln\left(\frac{1}{\delta}\right)}{\epsilon}$$

We can see that when deciding the sample size m sufficient for PAC learning, the size of hypothesis class $|\mathcal{H}|$ plays an important role. The larger the size of the hypothesis space, the more examples we need. We only know in our case the hypothesis class is finite, so why it is the case?

To approach this problem, I would like to start from a simplest example: labelling a sequence of numbers 1 to 30 as + or −. We may think of labelling the every prime number as +, or even labelling all odd number as −. In general, a hypothesis is a customized rule of labelling, and we are the rule maker. Before randomly sampling a training set, let's come up with 50 hypothesis.

Training each hypothesis on the training set, we can score the performance of each hypothesis based on the empirical error $\text{err}_S(h)$ on the training data. Based on the ERM rule, we will pick the hypothesis with the least error on the training set. Imagine that the set of all of our hypotheses was our hypothesis class $\mathcal{H}$ and that the our learning algorithm ERM has just output this least-error hypothesis which we will call h. If we test h to see how it preforms on test data that is labeled according to the same rule. It is possible that the h will have terrible performance on the test set.

The basic idea here is that we have so many hypotheses that one was bound to do very well; however, it did not generalize well to the test set because the test set is totally random, so we cannot expect it to generalize. This shows us that we should expect the performance getting worse as hypothesis classes grow larger; good performance of a hypothesis drawn from a large class on a training set may not tell us very much at all about how well it will generalize, unless we compensate for this larger class with more appropriately large amount of data. In other words, learning without restricting the hypothesis may result in overfitting.

As I was playing with this question, I found that it is indeed the justification for a fundamental theorem in the statistical learning theorem called "Occam's Razor". Now we know a smaller hypothesis class is usually better, so is there a way to concretely decide the cutoff of feasible hypothesis class size?

3.2 How small a hypothesis class should be?

The largest obstacles to answering this question would be understanding the sufficient and necessary condition for learning, which is commonly called the VC-dimension. Even though I did not cover the learnability of infinite class in this report, I think some examples of infinite-cardinality hypothesis classes will fully demonstrate the consequence of not restricting hypothesis class, such as the disjunction over n boolean variables. Vapnik and Chervonenkis discovered that a parameter the VC-dimension controls the learnability. To figure out in what way VC-dimension decides learnable hypothesis, we can explore the VC-dimension on both learnable and unlearnble examples in both finite and infinite hypothesis class:

- Positive half-lines
- Axis-aligned rectangles
- Convex polygons in Euclidean plane
- disjunctive formulas (DNF) (most important)
- finite classes

Comparing the these examples, we may discover what exactly characterizes what is learnable in the statistical (PAC) learning model for both finite and infinite hypothesis classes. While reading papers for this problem, I also notice there is another important concept we should pay attention along with the VC-dimension, which is sample complexity. Exploration on the relationship between VC-dimension, learnabilty, and sample complexity will be crucial steps for us to answer our question.

A Exercises

A.1 Exercise 1: Threshold Function

Consider we have a class of threshold functions $\mathcal{H} = 0.1, 0.2, ..., 0.9$, and let $x \in \mathbb{R}$ lies in the interval [0, 1]. For example, one of the members of $\mathcal{H}$ is the function:

$$h_{\theta=0.1}(x) = \begin{cases} 1 & \text{if } x > theta \\ 0 & \text{if } x < theta \end{cases}$$

1. what training set size we need for PAC learning for $\varepsilon = 0.01$ and
2. what if we increase ε

A.1.1 Solution for Exercise 1

1. From the section "Learning any function", we have $m \geq \frac{2}{\varepsilon^2}(\ln 2|\mathcal{H}| + \ln \frac{1}{delta})$. Plugging in each elements, we have $m \geq \dfrac{2}{0.01^2}() \ln 18 + ln\frac{1}{0.05} \approx 117722$.
2. Repeating the steps below, $m \geq \dfrac{2}{0.1^2}() \ln 18 + ln\frac{1}{0.05} \approx 117723$. We can observed that even though the ε become 10 times large, there isn't an obvious change in sufficient sample size.

A.2 Exercise 2: Understanding the Concept of PAC Learning

1. Please judge whether the statements below is true or false.
   (a) In a PAC learning model, the learner makes no assumptions about the class from which the target concept is drawn.
   (b) The number of training examples required for successful learning is strongly influenced by the complexity of the hypothesis space considered by the learner.
   (c) In PAC learning, the learner outputs the hypothesis from $\mathcal{H}$ that has the lowest error over the training data.
2. Suppose $\mathcal{H}$ is a finite hypotheses class and $\mathcal{S}$ is a set of training data. We would like our algorithm to output the most probable hypothesis $h \in \mathcal{H}$, given the data $\mathcal{S}$. Under what conditions does the following hold?
$$\text{argmax}_{h \in \mathcal{H}} P(h|\mathcal{D}) = \text{argmax}_{h \in \mathcal{H}} P(\mathcal{D}|h)$$

A.2.1 Solution for Exercise 2

1. True/False?
   (a) False
   (b) True
   (c) False
2. Conditions: $P(\mathcal{D})$ can be dropped because it does not depend on h $P(h)$ can be treated as a constant if all the hypotheses in the hypothesis space are equally likely.

A.3 Exercise 3: PAC Learning

After drop the realizability assumption that there is a hypothesis in H with zero true error, and move to agnostic PAC-learning. Let $\mathcal{H}$ be finite class. If we require

$$\mathbb{P}_{D\sim\mathbb{P}(X,Y)^m}\left[\text{err}_{\mathcal{D}}\left(h_D\right) > \min_{h\in\mathcal{H}}\left\{\text{err}_{\mathcal{D}}(h)\right\} + \epsilon\right] \leq \delta$$

where h_D is any hypothesis output by an ERM learner, then it suffices to obtain a sample that is $\frac{\varepsilon}{2}$ representative with probability at least $1-\delta$. That is, we need

$$\mathbb{P}_{D\sim\mathbb{P}(X,Y)^m}\left[\exists h\in\mathcal{H} : |\text{err}_{\mathcal{D}}(h) - \text{err}_D(h)| > \frac{\varepsilon}{2}\right] \leq \delta$$

By the union bound and Hoeffding's inequality we have that

$$\mathbb{P}_{D\sim\mathbb{P}(X,Y)^m}\left[\exists h\in\mathcal{H} : |\text{err}_{\mathcal{D}}(h) - \text{err}_D(h)| > \frac{\varepsilon}{2}\right] \leq 2|\mathcal{H}|e^{-\frac{1}{2}\varepsilon^2 m}$$

Hence, we need $\delta \geq 2|\mathcal{H}|e^{-\frac{1}{2}\varepsilon^2 m}$.

1. Derive a formula for the sufficient sample size to meet given (ε, δ) requirements. Compare this to the sample size that with the realizability assumption and explain the difference.

2. A data set D is called ε-representative w.r.t. domain Z, hypothesis class $\mathcal{H}$, and distribution $\mathcal{D}$ if

   $$\forall h\in\mathcal{H} : |\text{err}_D(h) - \text{err}_{\mathcal{D}}(h)| \leq \epsilon$$

   [6] Show that if the sample is 2ε representative with respect to $\mathcal{H}$, then $\text{err}(h_{\mathcal{D}}) \leq \min_{h\in\mathcal{H}} \text{err}_{\mathcal{D}}(h) + \varepsilon$, for any ERM hypothesis is $h_{\mathcal{D}}$.

A.3.1 Solution for Exercise 3

1. Starting from $\delta \geq 2|\mathcal{H}|e^{-0.5\varepsilon^2 m}$, we solve for m and get:

   $$m \geq \frac{2}{\varepsilon^2}\left(\ln 2|\mathcal{H}| + \ln\frac{1}{\delta}\right)$$

   For the one with realizability assumption,

   $$m \geq \frac{1}{\varepsilon}\left(\ln 1|\mathcal{H}| + \ln\frac{1}{\delta}\right)$$

   The most important difference is that in the agnostic case (without the realizability assumption) we have the factor $\frac{2}{\varepsilon^2}$ instead of 1 in the realizable case. Since $\varepsilon \in (0,1)$, and typically closer to 0, we have that ε^2 is smaller than ε, and hence $\frac{1}{\varepsilon^2}$ is bigger than $\frac{1}{\varepsilon}$. The rule is that we need more data in the agnostic case.

2. The idea of this question can be acquired from this image: The dots are the training error of different hypotheses. Since the sample is $2-\varepsilon$ representative, we know that the true error is within $2-\varepsilon$ of the training error. Our ERM-algorithm will return a hypothesis with minimum training error. In the picture, both the red and the blue hypothesis achieve the minimum training error, so an ERM algorithm choose either one of them. If our ERM algorithm returns the red one. Worst thing that can happen is that its true error (the red cross) is $2-\varepsilon$ higher than its training error, but for the blue hypothesis, the true error (the blue cross) is $2-\varepsilon$ smaller than its training error. However, the true

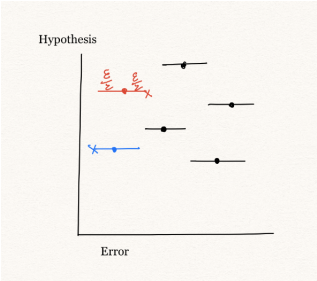


Figure 2: Agnostic PAC-learning

error of the selected hypothesis is still within ε of the best true error. Expressing in inequalities: for any $h \in \mathcal{H}$:

$$\begin{aligned} \mathrm{err}_D(h_D) &\leq \mathrm{err}_D(h_D) + \frac{\varepsilon}{2} \\ &\leq \mathrm{err}_D(h) + \frac{\varepsilon}{2} \\ &\leq \mathrm{err}_D(h) + \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\ &= \mathrm{err}_D(h) + \varepsilon \end{aligned}$$

Hence, on an $\frac{\varepsilon}{2}$-representative sample D, the $\mathrm{ERM}_{\mathcal{H}}$-rule yields an optimal result.

B Proof of Hoeffding's Inequality

[5]

Definition 4. *(Hoeffding's Inequality) Let $X_1, ..., X_m$ be iid random variables such that $a_i < X < b_i$ with probability 1. Let $S_n = \Sigma_{i=1}^n X_i$, then for any $t > 0$,*

$$P(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\Sigma_{i=1}^n (b_i - a_i^2)}}$$

Proof. The key to of this proof is the following upper bound: if X is a random variable with $E[X] = 0$ and $a \leq X \leq b$, then

$$E\left[e^{sX}\right] \leq e^{\frac{s^2(b-a)^2}{8}}$$

To derive the upper bound, by the convexity of the exponential function,

$$e^{sx} \leq \frac{x-a}{b-a}e^{sb} + \frac{b-x}{b-a}e^{sa}, \text{ for } a \leq x \leq b$$

Therefore,

$$\begin{aligned} E\left[e^{sX}\right] &\leq E\left[\frac{X-a}{b-a}\right]e^{sb} + E\left[\frac{b-X}{b-a}\right]e^{sa} \\ &= \frac{b}{b-a}e^{sa} - \frac{a}{b-a}e^{sb}, \text{ since } E[X] = 0 \\ &= \left(1 - \theta + \theta e^{s(b-a)}\right)e^{-\theta s(b-a)}, \text{ where } \theta = \frac{-a}{b-a} \end{aligned}$$

Now let

$$u = s(b-a) \text{ and define } \phi(u) \equiv -\theta u + \log\left(1 - \theta + \theta e^u\right)$$

Then we have

$$E\left[e^{sX}\right] \leq \left(1 - \theta + \theta e^{s(b-a)}\right)e^{-\theta s(b-a)} = e^{\phi(u)}$$

Using Taylor's expansion:

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(v) \text{ for some } v \in [0, u]$$

$$\begin{aligned} \phi'(u) &= -\theta + \frac{\theta e^u}{1 - \theta + \theta e^u} \Rightarrow \phi'(u) = 0 \\ \phi''(u) &= \frac{\theta e^u}{1 - \theta + \theta e^u}\left(1 - \frac{\theta e^u}{1 - \theta + \theta e^u}\right) \\ &= \rho(1 - \rho) \end{aligned}$$

Now, $\phi''(u)$ is maximized by

$$\rho = \frac{\theta e^u}{1 - \theta + \theta e^u} = \frac{1}{2} \Rightarrow \phi''(u) \leq \frac{1}{4}$$

So,

$$\begin{aligned} \phi(u) &\leq \frac{u^2}{8} = \frac{s^2(b-a)^2}{8} \\ \Rightarrow E\left[e^{sX}\right] &\leq e^{\frac{s^2(b-a)^2}{8}} \end{aligned}$$

Now, we can apply this upper bound to derive Hoeffding's inequality.

$$\begin{aligned}
P\left(S_n - E\left[S_n\right] \geq t\right) &\leq e^{-st} \prod_{i=1}^{n} E\left[e^{s(L_i - E[L_i])}\right] \\
&\leq e^{-st} \prod_{i=1}^{n} e^{\frac{s^2 (b_i - a_i)^2}{8}} \\
&= e^{-st} e^{s^2 \sum_{i=1}^{n} \frac{(b_i - a_i)^2}{8}} \\
&= e^{\frac{-2t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}} \\
&\text{by choosing } s = \frac{4t}{\sum_{i=1}^{n} \left(b_i - a_i\right)^2}
\end{aligned}$$

Similarly, $P\left(E\left[S_n\right] - S_n \geq t\right) \leq e^{\frac{-2t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}}$. This completes the proof of the Hoeffding's theorem. □

References

[1] Pranjal Awasthi. *CS 598: Theoretical Machine Learning: LECTURE 1*. Sept. 2017.

[2] Pranjal Awasthi. *CS 598: Theoretical Machine Learning: LECTURE 3*. Sept. 2017.

[3] Elad Hazen. *THEORETICAL MACHINE LEARNING (COS511: LECTURE 1*. Feb. 2015.

[4] Percy Liang. *CS229T/STAT231: Statistical Learning Theory (Winter 2016)*.

[5] Robert D. Nowak. *ECE901 Spring 2007 Statistical Learning Theory: Lecture 7: Chernoff's Bound and Hoeffding's Inequality*. 2007.

[6] Arno Siebes. *PAC Learning and the VC Dimension*.