Chapter ONE Probably Approximately Correct (PAC)

Siheng Zhang
zhangsiheng@cvte.com

September 6, 2020

The notes is mainly based on the following books:

- Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz and Shai Ben-David, 2014 [1]
- pattern recognition and machine learning, Christopher M. Bishop, 2006 [2]
- Probabilistic Graphical Models: Principles and Techniques, Daphne Koller and Nir Friedman, 2009 [3]
- Graphical Models, Exponential Families, and Variational Inference, Martin J. Wainwright and Michael I. Jordan, 2008 [4]

This part corresponds to **Chapter 2-5 in UML**, and mainly answers the following questions:

- What can we know about the generalization error?
- How does the hypothesis set (in application, the choice of classifier/regressor or so on) reflect our prior knowledge, or, inductive bias?

[1] https://www.cs.huji.ac.il/s̃hais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf
[2] http://users.isr.ist.utl.pt/w̃urmd/Livros/school/Bishop - Pattern Recognition And Machine Learning - Springer 2006.pdf
[3] https://mitpress.mit.edu/books/probabilistic-graphical-models
[4] https://people.eecs.berkeley.edu/w̃ainwrig/Papers/WaiJor08_FTML.pdf

Contents

1 Formulation

1.1 The learner's input, output, and evaluation

- **input**:
  - Domain set: instance $x \in \mathcal{X}$.
  - Label set: label $y \in \mathcal{Y}$. Currently, just consider the binary classification task.
  - Training set: $S = ((x_1, y_1), \cdots, (x_m, y_m))$ is a finite sequence.
- **output**: hypothesis (or classifier, regressor) $h : \mathcal{X} \to \mathcal{Y}$.
- **data generation model**: Assume that the instances are generated by some probability distribution $\mathcal{D}$, and there is some 'correct' labeling function (currently): $f : \mathcal{X} \to \mathcal{Y}$.

  The i.i.d. assumption: the training samples are independently and identically distributed.

  remark1: The learner is blind to the data generation model.

  remark2: Usually called 'training set', but must be 'training sequence', because the same samples may repeat, and some training algorithms are order-sensitive.

  remark3: Strictly speaking, the distribution $\mathcal{D}$ is defined over $\mathcal{X} \times \mathcal{Y}$.
- **Generalization error**: *a.k.a*, true error/risk.

$$L_{\mathcal{D},f}(h) \stackrel{def}{=} \mathop{\mathbb{P}}_{x\sim\mathcal{D}}[h(x) \neq f(x)] \stackrel{def}{=} \mathcal{D}(x : h(x) \neq f(x)) \tag{1}$$

2 From ERM to PAC

2.1 ERM (Empirical Risk Minimization) may lead to overfitting

Since the generalization error is intractable, turn to minimize the **empirical risk**:

$$L_S(h) \stackrel{def}{=} \frac{|\{(x_i, y_i) \in S : h(x_i) \neq y_i\}|}{m} \tag{2}$$

Consider a 'lazy' learner h, which predict $y = y_i$ iff. $x = x_i$, and 0 otherwise. It has 1/2 probability to fail for unseen instances, i.e., $L_{\mathcal{D},f}(h) = 1/2$, while $L_S(h) = 0$. Hence, it is an excellent learner on the training set, but a poor learner in the universe case. This phenomenon is called 'overfitting'. The lesson behind this learner is: without restriction on the hypothesis set, ERM can lead to overfitting.

2.2 ERM with restricted hypothesis set (inductive bias)

Instead of $h_S \in \arg\min L_S(h)$, ERM with restricted hypothesis set return the following hypothesis:

$$h_S \in \arg\min_{h\in\mathcal{H}} L_S(h) \tag{3}$$

Start from an ideal case, in which the **realizability assumption** holds, i.e., there exists $h^* \in \mathcal{H}$, such that $L_{\mathcal{D},f}(h^*) = 0$.

It implies that $L_S(h^*) = 0$, $L_S(h_S) = 0$. However, we are interested in $L_{\mathcal{D},f}(h_S)$.

2.3 PAC (Probably Approximately Correct) learnability

Definition: Training on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ samples, there exists an algorithm to be able to achieve **accuracy** at least $1 - \epsilon$ with **confidence** at least $1 - \delta$.

Theorem 1 *Finite hypothesis classes are PAC learnable, and the sample complexity is:* $m_{\mathcal{H}}(\epsilon, \delta) = \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$.

Proof *Let $\mathcal{H}_B$ be the set of 'bad' hypothesis, that is, $\mathcal{H}_B \subset \mathcal{H}$, and $\forall h \in \mathcal{H}_B, L_{\mathcal{D},f}(h) > \epsilon$. Let M be the set of 'misleading' samples, that is $M = \{S : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$. Note that,*

$$M = \bigcup_{h \in \mathcal{H}_B} \{S : L_S(h) = 0\}$$

The goal is to bound the probability of the event $L_{\mathcal{D},f}(h_S) > \epsilon$,

$$\begin{aligned}
&\mathcal{D}^m(\{S : L_{\mathcal{D},f}(h_S) > \epsilon\}) \leq \mathcal{D}^m(M) \\
&= \mathcal{D}^m(\bigcup_{h \in \mathcal{H}_B} \{S : L_S(h) = 0\}) = \sum_{h \in \mathcal{H}_B} \prod_{i=1}^{m} \mathcal{D}(\{x_i : f(x_i) = h(x_i)\}) \\
&\overset{i.i.d.}{=} \sum_{h \in \mathcal{H}_B} (1 - L_{\mathcal{D},f}(h))^m \leq \sum_{h \in \mathcal{H}_B} (1 - \epsilon)^m \leq \sum_{h \in \mathcal{H}_B} \exp(-\epsilon m) \\
&\leq |\mathcal{H}| \exp(-\epsilon m)
\end{aligned}$$

Let $|\mathcal{H}| \exp(-\epsilon m) \leq \delta$, we can solve that $m \geq \log(|\mathcal{H}|/\delta)/\epsilon$.

2.4 No-Free-Lunch

Theorem 2 *Let A be any learning algorithm for the task of binary classification with respect to the 0-1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $\mathcal{X}/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $X \times \{0, 1\}$ such that:*

- *There exists a function $f : \mathcal{X} \to \{0, 1\}$ with $L_{\mathcal{D}}(f) = 0$.*
- *With probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$.*

Proof *Let $C \subseteq \mathcal{X}$ of size $2m$. There are $T = 2^{2m}$ possible functions $f_1, \cdots, f_T$ defined on $C \to \{0, 1\}$. For each such function, let $\mathcal{D}_i$ be a distribution over $\mathcal{C} \times \{0, 1\}$ defined as follow:*

$$\mathcal{D}_i(\{(x, y)\}) = \begin{cases} 1/|C|, & \text{if } y = f_i(x) \\ 0, & \text{otherwise} \end{cases}$$

There are $k = (2m)^m$ possible sequences $S_1, \cdots, S_k$, each of m examples from C. Also, we denote the sequence $\mathcal{S}_j$ labelled by the function f_i as $\mathcal{S}_j^i$. If the distribution is $\mathcal{D}_i$, then the possible training sets are $S_1^i, \cdots, S_k^i$, with equal probability. Therefore,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}_i^m}[L_{\mathcal{D}_i}(A(S))] = \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i))$$

Using the facts that 'maximum' is larger than 'average', and that 'average' is larger than 'minimum', we have

$$\max_{i \in \{1, \cdots, T\}} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) \geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(A(S_j^i)) \geq \min_{j \in \{1, \cdots, k\}} \frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i))$$

Next, fix some $j \in \{1, \cdots, k\}$. Denote $S_j = (x_1, \cdots, x_m)$ and let $v_1, \cdots, v_p$ be the examples in C that do not appear in S_j. Clearly, $p \geq m$. Therefore, for every $h : C \to \{0, 1\}$,

$$L_{\mathcal{D}_i}(h) = \frac{1}{2m} \sum_{x \in C} \mathbb{I}_{[h(x) \neq f_i(x)]} \geq \frac{1}{2p} \sum_{r=1}^{p} \mathbb{I}_{[h(v_r) \neq f_i(v_r)]}$$

and hence,

$$\frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)) \geq \frac{1}{2} \min_{r \in \{1, \cdots, p\}} \frac{1}{T} \mathbb{I}_{[A(S_j^i)(v_r) \neq f_i(v_r)]}$$

Next, fix some $r \in [p]$, We can partition all the functions $f_1, \cdots, f_T$ into $T/2$ disjoint pairs, where for a pair $(f_i, f_{i'})$ we have that $\forall c \in C, f_i(c) \neq f_{i'}(c)$ iff. $c = v_r$. Since for such a pair we must have $S_j^i = S_j^{i'}$, it follows that $\mathbb{I}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} + \mathbb{I}_{[A(S_j^{i'})(v_r) \neq f_{i'}(v_r)]} = 1$, which yields:

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{I}_{[A(S_j^i)(v_r)\neq f_i(v_r)]}=\frac{1}{2}$$

In conclusion, it holds that

$$\max_{i\in\{1,\cdots,T\}}\mathop{\mathbb{E}}_{S\sim\mathcal{D}_i^m}[L_{\mathcal{D}_i}(A(S))]\geq 1/4$$

This means that for every algorithm, there exists $f,\mathcal{D}$, such that $L_{\mathcal{D}}(f)=0$ and $\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))]\geq 1/4$.

(following is part of UML Ex5.1) For a random variable $\theta\in[0,1]$ such that $\mathbb{E}(\theta)\geq 1/4$, we have:

$$p\left(\theta\geq\frac{1}{8}\right)=\int_{\frac{1}{8}}^{1}p(\theta)\mathrm{d}\theta\geq\int_{\frac{1}{8}}^{1}\theta p(\theta)\mathrm{d}\theta=\mathbb{E}(\theta)-\int_{0}^{\frac{1}{8}}\theta p(\theta)\mathrm{d}\theta\geq\mathbb{E}(\theta)-\frac{1}{8}\int_{0}^{\frac{1}{8}}p(\theta)\mathrm{d}\theta=\frac{1}{4}-\frac{1}{8}\left(1-\int_{\frac{1}{8}}^{1}p(\theta)\mathrm{d}\theta\right)$$

which leads to $p(\theta\geq 1/8)\geq 1/7$.

NFL theorem tells the neccessity of inductive bias. Philosophically, if someone can explain every phenomenon, his explanations are worthless.

2.5 Agnostic PAC

2.5.1 Beyond realizability assumption

In practical, the 'true' labelling function may not exist, and the labels may not be fully determined by the features on hand. Then Agnostic PAC learnability is defined as: training on $m\geq m_{\mathcal{H}}(\epsilon,\delta)$ samples, there exists an algorithm with **confidence** at least $1-\delta$ to achieve that:

$$L_{\mathcal{D}}(h)\leq\min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\epsilon$$

in which $L_{\mathcal{D}}(h)\overset{def}{=}\mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x)\neq y]\overset{def}{=}\mathcal{D}(\{x:h(x)\neq y\})$.

2.5.2 Beyond binary classification

Agnostic PAC learnability remains the same with:

$$\mathcal{D}(h)=\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[l(h,z)]\tag{4}$$

in which $l(\cdot)$ is 0-1 loss for multiclass classification and square loss for regression.

2.5.3 Sample complexity under Agn-PAC: via uniform convergence

Definition ϵ-representative: A training set S is called ϵ-representative if $\forall h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|\leq\epsilon$.

Theorem 3 *ERM rule is suitable for $\epsilon/2$-representative samples.*

Proof *for every $h\in\mathcal{H}$,*

$$L_{\mathcal{D}}(h_S)\leq L_S(h)+\frac{\epsilon}{2}\leq L_S(h)+\frac{\epsilon}{2}\leq L_S(h)+\epsilon\tag{5}$$

Hence, $L_{\mathcal{D}}(h_S)\leq\min_{h\in\mathcal{H}}L_S(h)+\epsilon$.

Theorem 4 ***Agnostic PAC sample complexity*** *Assume that the range of the loss function is ¡$[0,1]$, or more general, $[a,b]$, then a finite hypothesis set $\mathcal{H}$ enjoys the agnostic PAC learnability with sample complexity*

$$m_{\mathcal{H}}(\epsilon,\delta)\leq m_{\mathcal{H}}^{UC}(\epsilon/2,\delta)\leq\left\lceil\frac{2\log(2|\mathcal{H}|/\delta)(b-a)^2}{\epsilon^2}\right\rceil\tag{6}$$

3 Error decomposition

$$\begin{aligned} L_{\mathcal{D}}(h_S) &= \epsilon_{\text{app}} + \epsilon_{\text{est}} \\ \epsilon_{\text{app}} &= \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \\ \epsilon_{\text{est}} &= L_{\mathcal{D}}(h_S) - \epsilon_{\text{app}} \end{aligned} \tag{7}$$

- **The Approximation Error** measures how much risk we have because we restrict ourselves to a specific class, namely, how much *inductive bias* we have. The approximation error does not depend on the sample size and is determined by the hypothesis class chosen. Enlarging the hypothesis class can decrease the approximation error.
- **The Estimation Error** measures the empirical risk (i.e., training error), which is only an estimate of the true risk. The quality of this estimation depends on the training set size (decreases with it) and on the size, or complexity, of the hypothesis class (logarithmically increases with it).

4 Summary

Now that, we have come to some important conclusions under the PAC learning framework:

1. No universal learner;
2. Inductive bias is neccessary to avoid overfitting;
3. Sample complexity is function about hypothesis set, confidence level and error, interestingly, it is nothing to do with the dimension of feature space;
4. Inductive bias controls the balance of approximation error and estimation error.

We have reached the fundamental question in learning theory: **Over which hypothesis classes, ERM learning will not result in overfitting (or, PAC learnable)?** Currently, we just confirm the PAC learnability for finite classes. In the next chapter, the most important part in learning theory, VC-dimension, will gives a more precise answer.

5 Excercises and solutions

Ex1 (UML Ex2.2) Let $\mathcal{H}$ be a class of binary classifiers over a domain $\mathcal{X}$. Let $\mathcal{D}$ be an unknown distribution over $\mathcal{X}$, and let f be the target hypothesis in $\mathcal{H}$. Fix some $h \in \mathcal{H}$, show that the expected value of $L_S(h)$ over the choice of S equals $L_{\mathcal{D},f}(h)$, namely,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_S(h)] = L_{\mathcal{D},f}(h)$$

Solution: according to the definition,

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_S(h)] &= \sum_S \mathcal{D}^m(S) \frac{|\{(x_i, y_i) \in S : h(x_i) \neq y_i\}|}{m} \\ &= \sum_S \mathcal{D}\{(x_i, y_i) \in S : h(x_i) \neq y_i\} \\ &= \mathcal{D}(\{x : h(x) \neq f(x)\}) = L_{\mathcal{D},f}(h) \end{aligned}$$

Ex2 (UML Ex2.3) **Axis Aligned rectangles**: An axis aligned rectangle classifier in the plane is a classifier that assigns the value 1 to a point if and only if it is inside a certain rectangle. Formally, given real numbers $a_1 \leq b_1, a_2 \leq b_2$, define the classifier $h(a_1, b_1, a_2, b_2)$ by:

$$h(a_1, b_1, a_2, b_2)(x_1, x_2) = \begin{cases} 1, & \text{if } a_1 \leq x_1 \leq b1 \text{ and } a_2 \leq x_2 \leq b_2 \\ 0, & \text{otherwise} \end{cases}$$

The class of all axis aligned rectangles in the plane is defined as:

$$\mathcal{H}^2_{\text{rec}} = \{h_{(a_1,b_1,a_2,b_2)} : a_1 \leq b_1, \text{ and } a_2 \leq b_2\}$$

Note that this is an infinite size hypothesis class. Throughout this exercise we rely on the realizability assumption.

2.1 Let A be the algorithm that returns the smallest rectangle enclosing all positive examples in the training set. Show that A is an ERM.

Ex3 (UML Ex3.2) Let $\mathcal{X}$ be a discrete domain, and let $\mathcal{H}_{\text{Singleton}} = \{h_z : z \in \mathcal{X}\} \cup \{h^-\}$, where for each $z \in \mathcal{X}$, h_z is the function defined by $h_z(x) = 1$ if $x = z$ and $h_z(x) = 0$ if $x \neq z$. h^- is simply the all-negative hypothesis, namely, $\forall x \in \mathcal{X}, h^-(x) = 0$. The realizability assumption here implies that the true hypothesis f labels negatively all examples in the domain, perhaps except one.

3.1 Describe an algorithm that implements the ERM rule for learning $\mathcal{H}_{\text{Singleton}}$ in the realizable setup.

3.2 Show that $\mathcal{H}_{\text{Singleton}}$ is PAC learnable. Provide an upper bound on the sample complexity.

Solution:

3.1 Traverse $z \in \mathcal{X}$ then output h_z or h^-.

3.2 If for any $i \in [1, \cdots, m]$, h_{x_i} is the true hypothesis, the algorithm can find it in the realizable setup. Otherwise, the algorithm outputs h^-, which can be either true or false (i.e., the target z^* is not in the training set). Note that in the second case, the algorithm only makes a single error when generalize to all cases, and hence $p(z^*) \geq \epsilon$ (otherwise, it is meaningless),

$$\mathbb{P}(L_{\mathcal{D},f}(h_S) > \epsilon) \leq (1 - p(z^*))^m \leq (1 - \epsilon)^m \leq \exp(-\epsilon m) \leq \delta$$

which leads to

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(1/\delta)}{\epsilon} \right\rceil$$

Ex4 (UML Ex3.3) Let $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{0, 1\}$, and let $\mathcal{H}$ be the class of concentric circles in the plane, that is, $\mathcal{H} = \{h_r : r \in \mathbb{R}_+\}$, where $h_r(x) = \mathbb{I}_{[\|x\| \leq r]}$. Prove that $\mathcal{H}$ is PAC learnable (assume realizability), and its sample complexity is bounded by

Ex5 (UML Ex3.4)

Ex6 (UML Ex3.7) The Bayes optimal predictor: Show that for every probability distribution $\mathcal{D}$, the Bayes optimal predictor $f_{\mathcal{D}}$ is optimal, in the sense that for every classifier g from X to $\{0, 1\}$, $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

Solution: The Bayes predictor labels a sample x according to

$$f_{\mathcal{D}}(x) = \begin{cases} 0, & \text{if } \mathcal{D}((x,0)) \geq \mathcal{D}((x,1)) \\ 1, & \text{otherwise} \end{cases}$$

When it labels a sample to be class 0, it holds that $\mathcal{D}((x,0)) \geq \mathcal{D}((x,1))$. If the true label function also makes such a decision, then Bayes predictor makes no error. Otherwise, $f(x) = 1$, but its probability is no more than $1/2$. Any other classifier that labels x to be class 1 will suffer a risk no less than $1/2$. Hence, in total, Bayes predictor is the optimal.

Ex7 (UML Ex3.9) Consider a variant of the PAC model in which there are two example oracles: one that generates positive examples and one that generates negative examples, both according to the underlying distribution $\mathcal{D} on \mathcal{X}$. Formally, given a target function $f : \mathcal{X} \to \{0, 1\}$, let $\mathcal{D}^+ bethedistributionover \mathcal{X}^+ = \{x \in \mathcal{X} : f(x) = 1\}$ defined by $\mathcal{D}^+(A) = \mathcal{D}(A)/\mathcal{D}(\mathcal{X}^+)$, for every $A \in \mathcal{X}^+$. Similarly, $\mathcal{D}^-$ is the distribution over $\mathcal{X}^-$ induced by $\mathcal{D}$.

The definition of PAC learnability in the two-oracle model is the same as the standard definition of PAC learnability except that here the learner has access to $m^+_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples from $\mathcal{D}^+$ and $m^-(\epsilon, \delta)$ i.i.d. examples from $\mathcal{D}^-$. The learner's goal is to output h s.t. with probability at least $1 - \delta$ (over the choice of the two training sets, and possibly over the nondeterministic decisions made by the learning algorithm), both $L(\mathcal{D}^+, f)(h) \leq \epsilon$ and $L(\mathcal{D}^-, f)(h) \leq \epsilon$.

7.1 Show that if H is PAC learnable (in the standard one-oracle model), then H is PAC learnable in the two-oracle model.

7.2 Define h^+ to be the always-plus hypothesis and h^- to be the always minus hypothesis. Assume that $h^+, h^- \in \mathcal{H}$. Show that if $\mathcal{H}$ is PAC learnable in the two-oracle model, then $\mathcal{H}$ is PAC learnable in the standard one-oracle model.

Ex8 (UML Ex5.3) Prove that if $|\mathcal{X}| \geq km$ for a positive integer $k \geq 2$, then we can replace the lower bound in the No-Free-Lunch theorem. Namely, for the task of binary classification, there exists a distribution $\mathcal{D} \sim \mathcal{X} \times \{0, 1\}$ such that:

8.1 There exists a function $f : \mathcal{X} \to \{0, 1\}$ with $L_{\mathcal{D}}(f) = 0$.

8.2 $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \frac{1}{2} - \frac{1}{2k}$.

Solution: Only the second proposition should be proved. Similar with the proof in above,

$$L_{\mathcal{D}_i}(h) = \frac{1}{km}\sum_{x\in C}\mathbb{I}_{[h(x)\neq f_i(x)]} \geq \frac{1}{km}\sum_{r=1}^{p}\mathbb{I}_{[h(v_r)\neq f_i(v_r)]} \geq \frac{k-1}{pk}\sum_{r=1}^{p}\mathbb{I}_{[h(v_r)\neq f_i(v_r)]}$$

And similarity,

$$\frac{1}{T}\sum_{i=1}^{T}L_{\mathcal{D}_i}(A(S_j^i)) \geq \frac{k-1}{k}\min_{r\in\{1,\cdots,p\}}\frac{1}{T}\mathbb{I}_{[A(S_j^i)(v_r)\neq f_i(v_r)]}$$

So the final bound is $1/2 - 1/2k$.

To be continue...
Chapter 2. VC-dimension
Chapter 3. Bayesian-PAC
Chapter 4. Generalization in Deep Learning