Topic 1 **July 18,2017**

Learnability

About: Machine Learning **Author: Yi-Shan Wu**

This note is written according to Understanding Machine Learning: From Theory to Algorithm. This week, we are going to introduce PAC learnability, which is an important model that help us figure out whether there is any restriction for good learning.

1 Recap

There are some basic elements in a learning model.

- Data distribution D (unknown)
- Target function f (unknown)
- Learning algorithm A
- Hypothesis set H
- Training datas S, datas in S are sampled from D

In general, the goal of a learning algorithm A is to output a function g chosen from H after seeing some training datas S, such that function g can be as similar as f.

2 Types of Learning

2.1 Learning paradigms

Here we list some useful learning paradigms. Different paradigms are used to deal with different situations and different assumptions. The data sets of the first three types of learning are given to us in "batch". That is, they are given entirely at the outset of the learning process.

- **Supervised Learning:** In supervised learning, each data is a pair consisting of input x and output y. All the discussions in this lecture is under this paradigm.
- **Unsupervised Learning:** The learner is just given input examples x without output y. The learner still knows what kind of output he is supposed to output. For example, $y \in \{0, 1\}$ for binary classification.
- **Reinforcement Learning:** The training datas are of the form $(input, someoutput, gradeofoutput)$ instead of $(input, correctoutput)$ as supervised learning. Thus, the learner is supposed to learn the best strategy according to those grades. This kind of learning paradigm is especially useful for learning how to play a game.

- **Online learning:** In online learning, datas are given one example at a time. That is, in each round, the algorithm follow the steps:
  1. Receive x_t
  2. Predict $\hat{y_t}$ according to hypothesis h_t
  3. Receive true label y_t
  4. Update h_t according some loss $l(\hat{y_t}, y_t)$

2.2 Labels of Y

- binary classification: $y = \{0, 1\}$ or $y = \{+1, -1\}$
- multi-class classification: $y \in \{1, 2, \cdots k\}$
- regression: $y \in \mathbb{R}$

3 PAC learnability

3.1 identical independent distribution

Suppose we receive m training samples $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$. y is labeled by some target function f, such that for every samples, $y = f(x)$. The goal of learning is to find out the target function f. Since we don't know the data distribution D, the only way we can access data is through training samples S. Thus, it is natural that we hope the data in S can represent D well.

First assumption: (i.i.d assumption)

Every data x is identical independent distribution (i.i.d.) from D. It can be denoted by $x \sim D$. If there are m samples, then it can be written as $S \sim D^m$.

3.2 Measure of success

- true error: $L_{D,f}(h) := \Pr_{x \sim D}[h(x) \neq f(x)]$
- training error: $L_S(h) := \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$

The goal of a learner is supposed to minimize true error $L_{D,f}(h)$ by some methods. ERM paradigm may be wrong (overfitting) since it simply minimize $L_S(h)$ without considering $L_{D,f}(h)$. However, instead of just throw away ERM paradigm, we try to fix it by some restrictions.
First, we restrict the selections of h that h can only be chosen from a hypothesis set H. Therefore, the output of ERM paradigm, h_S now becomes

$$h_S = ERM_H(S) \in argmin_{h \in H} L_S(h)$$

Later we will show that if H is finite, then the hypothesis set H may be learnable.

3.3 realizable assumption

To make things easy, here we assume that the target function $f = h^* \in H$. That is, $\exists h^* \in H$, such that $L_{D,f}(h^*) = 0$. This is called "realizable assumption". (It is a very strong assumption that even seems to be unreasonable. We will try to release this assumption next section.)

3.4 bound of bad samples

Since there is "probability" due to the i.i.d. sampling process, we estimate "success" of learning by parameters ϵ and δ:

$$\Pr[L_{D,f}(h_S) \leq \epsilon] \geq 1 - \delta$$

Equivalently, we want to show

$$\Pr[L_{D,f}(h_S) > \epsilon] \leq \delta$$

If the output of a learner h_S has the probability greater than $(1 - \delta)$ that true error will be less than ϵ, then we say that the hypothesis set H is learnable by ERM paradigm.
However, where does ϵ comes from? Recall that the output h_S is decided deterministic according to some training samples S. Thus, a "bad" sample may lead to misleading the algorithm. To bound the probability of error, we have to bound the probability to sample these "bad" samples. That is to bound

$$D^m(M') = D^m(\{S : L_{D,f}(h_S) > \epsilon, L_S(h_S) = 0\}) \tag{1}$$

where $D(S : condition = true) := \Pr_{S \sim D}[condition]$.
Intuitively, it is the probability of under realizability assumption, (causes $L_S(h_S) = 0$, where h_S is the chosen hypothesis by ERM paradigm) that the algorithm being mislead by sample S.
Next, we define H_B to be the set that contains all the hypotheses that are "truly bad". That is,

$$H_B : \{h \in H : L_{D,f}(h) > \epsilon\} \tag{2}$$

Note that equation (1) contains the samples that make learning "fail" while equation (2) contains the hypothesis that make learning "fail". We can construct a table with V denotes those who make learning "fail".

	h_1	h_2	h_3	h_4	h_5
S_1		V		V	
S_2	V	V		V	
S_3		V	V		V
S_4		V		V	
S_5					

Note that the goal here is to bound M'. Unfortunately, we are not able to get a bound directly. Thus, we turn to find the bound by introducing another released set M, which is defined as

$$M : \{S : \exists h \in H_B, L_S(h) = 0\} = \bigcup_{h \in H_B} \{S : L_S(h) = 0\} \tag{3}$$

To compare equation (1) and equation (3), it is obvious that $M' \subseteq M$. The relationship also implies $D^m(M') \leq D^m(M)$. Therefore,

$$\begin{aligned}
&D^m(\{S : L_{D,f}(h_S) > \epsilon, L_S(h_S) = 0\}) \overset{(a)}{\leq} D^m(\bigcup_{h \in H_B} \{S : L_S(h) = 0\}) \\
&\overset{(b)}{\leq} \sum_{h \in H_B} D^m(S : L_S(h) = 0) = \sum_{h \in H_B} D^m(S : \forall i, h(x_i) = f(x_i)) \\
&\overset{(c)}{=} \sum_{h \in H_B} \Pi_{i=1}^m D(x_i : h(x_i) = f(x_i)) \\
&= \sum_{h \in H_B} \Pi_{i=1}^m (1 - L_{D,f}(h)) \leq \sum_{h \in H_B} \Pi_{i=1}^m (1 - \epsilon) \leq \sum_{h \in H_B} e^{-\epsilon m} \\
&\leq |H_B| e^{-\epsilon m} \leq |H| e^{-\epsilon m}
\end{aligned}$$

where, $\overset{(a)}{\leq}$ comes from $M' \subseteq M$, $\overset{(b)}{\leq}$ comes from "union bound" and $\overset{(c)}{\leq}$ comes from "i.i.d. assumption".

3.5 sample complexity

Note that to guarantee $|H|e^{-\epsilon m} \leq \delta$, we must have

$$m \geq \frac{log(|H|/\delta)}{\epsilon}$$

We define $m_H \leq \lceil \frac{log(|H|/\delta)}{\epsilon} \rceil$ as "sample complexity", which is the lower bound of the number of samples to learn hypothesis H. Then, hypothesis H seems to be learnable if $|H|$ is finite and with sample size greater than m_H.

3.6 PAC learnability

Now, we can define PAC learnability.

Definition 1 *(PAC learnability)*
A hypothesis class H is PAC learnable if there is an algorithm A and a function $m_H : (0,1)^2 \to \mathbb{N}$ such that $\forall \epsilon, \delta \in (0,1)$ and for every D over X and a target function $f : X \to \{0,1\}$, and the realizable assumption hold, then when running A with $m \geq m_H(\epsilon, \delta)$ i.i.d. examples generated from D and labeled by f, the algorithm return a h_S such that

$$\Pr_{S \sim D^m}[L_{D,f}(h_S) \leq \epsilon] \geq 1 - \delta$$

4 Agnostic PAC learnability

Release the realizable assumption

In the previous section, we have an assumption that seems to be not practical: **realizable assumption**. First of all, we never know whether the target function is in our hypothesis class H or

not. Second, in real world, there are usually noises in out samples. y may not able to be written as $f(x)$. Thus, from now on, let D be a probability distribution over $X \times Y$. That is, D is a joint distribution over domain points and labels.

Redefine error

- true error: $L_D(h) := \Pr_{(x,y)\sim D}[h(x) \neq f(x)]$
- training error: $L_S(h) := \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$ (the same as before)

The goal now is to find a h to minimize the true error $L_D(h)$. With these changes we define another definition of learnability:

Definition 2 *(Agnostic PAC learnability) A hypothesis class H is agnostic PAC learnable if there is an algorithm A and a function $m_H : (0,1)^2 \to \mathbb{N}$ such that $\forall \epsilon, \delta \in (0,1)$ and for every D over $X \times Y$, then run A with $m \geq m_H(\epsilon, \delta)$ i.i.d. examples generated from D , the algorithm return a h_S such that*

$$\Pr_{S\sim D^m}[L_{D,f}(h_S) \leq min_{h' \in H} L_D(h') + \epsilon] \geq 1 - \delta$$

Clearly, if the realizability assumption holds, the agnostic PAC learnability provides the same guarantee as PAC learning. In that sense, agnostic PAC learning generalizes the definition of PAC learning.

5 Uniform Convergence Property

Here we are going to release another setting. In previous sections, we wish that learning algorithm can find some h such that $L_S(h) = 0$. Frankly speaking, to simply find a h such that $L_S(h)$ is small isn't quite rational. It is rather reasonable to hope $L_S(h)$ performs similar to $L_D(h)$ for every hypothesis in H such that we can approximate true error, $L_D(h)$, according to $L_S(h)$. Therefore, we are going to find out the condition to guarantee that $L_S(h)$ is close to $L_D(h)$.

Definition 3 *(ϵ-representative sample)*
A training sample is called ϵ-representative if

$$\forall h \in H, |L_S(h) - L_D(h)| \leq \epsilon$$

This definition seems to be really "strong" since it require that the ϵ-representative sample S satisfy every hypothesis in the hypothesis class. And for this reason, this property is said to be "uniform" for all $h \in H$. The next lemma shows that whenever the sample is $(\epsilon/2)$-representative, the ERM algorithm is guaranteed to return a good hypothesis.

Lemma 4 *Assume the sample S is $(\epsilon/2)$-representative, then the output of ERM algorithm, $h_S \in ERM_H(S)$ satisfies,*

$$L_D(h_S) \leq min_{h' \in H} L_D(h') + \epsilon$$

Thus, it is obvious that to ensure agnostic PAC learnability, it suffices to ensure that with probability greater than $(1 - \delta)$ that the sample $S \sim D$ is $(\epsilon/2)$-representative. In the following subsection, we are going to figure out the sample complexity ($m_H^{UC}(\epsilon/2, \delta)$) suffices for $\epsilon/2$-representative sample and the relationship between $m_H(\epsilon, \delta)$ and $m_H^{UC}(\epsilon, \delta)$.

5.1 UC property v.s. learnability

To ensure that with probability greater than $(1-\delta)$ that the sample $S \sim D$ is $(\epsilon/2)$-representative, we can fix some ϵ, δ and find a sample size m such that for any D,

$$D^m(\{S : \forall h \in H, |L_S(h) - L_D(h)| \leq \epsilon/2\}) \geq 1 - \delta$$

Note that $\forall h \in H$ is contained inside the probability since the explanation for "uniform" mentioned previously.
Equivalently, we need to show,

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon/2\}) \leq \delta$$

Again, we can use union bound and get,

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon/2\}) \leq \sum_{h \in H} D^m(\{S : |L_S(h) - L_D(h)| > \epsilon/2\}) \tag{4}$$

Since in general,

$$L_S(h) = \frac{\Sigma_{i=1}^m l(h, z_i)}{m}$$
$$L_D(h) = \mathop{\mathrm{E}}_{z \sim D}[l(h, z)]$$

and every l for a single sample are i.i.d. We can use *Hoeffding's inequality* to bound equation (4)

$$(4) = \sum_{h \in H} \Pr_{S \sim D^m}[|L_S(h) - L_D(h)| > \epsilon/2] \leq \sum_{h \in H} 2exp(-m\epsilon^2/2) = 2|H|exp(-m\epsilon^2/2)$$

To ensure $2|H|exp(-m\epsilon^2/2) \leq \delta$, we have,

$$m \geq \frac{2log(2|H|/\delta)}{\epsilon^2}$$

We can also define the sample complexity to ensure the high probability to sample $\epsilon/2$-representative sample to be

$$m_H^{UC}(\epsilon/2, \delta) \leq \lceil \frac{2log(2|H|/\delta}{\epsilon^2} \rceil$$

Here we come back to define the "Uniform Convergence" property. The term "uniform" here refers to having a fixed sample size that works for all $h \in H$ and all D over Z.

Definition 5 *(Uniform Convergence property)*
A hypothesis class H has UC property if there exists a function $m_H^{UC} : (0,1)^2 \to \mathbb{N}$ such that $\forall \epsilon, \delta \in (0,1)$ and for every D over Z, if S is a sample of $m \geq m_H^{UC}(\epsilon, \delta)$ examples drawn i.i.d. from D, then,

$$\Pr_{S \sim D^m}[S \text{ is } \epsilon - representative] \geq 1 - \delta$$

Corollary 6 *Let H be a finite hypothesis class, let Z be a domain, and let $l : H \times Z \to [0,1]$. Then, H enjoys UC property with sample complexity*

$$m_H^{UC}(\epsilon, \delta) \leq \lceil \frac{log(2|H|/\delta}{2\epsilon^2} \rceil$$

Therefore, we can conclude that if H has UC property with sample complexity $m_H^{UC}(\epsilon/2, \delta)$, the class is agnostic PAC learnable under ERM paradigm with sample complexity

$$m_H(\epsilon, \delta) \leq m_H^{UC}(\epsilon/2, \delta) \leq \lceil \frac{2log(2|H|/\delta}{\epsilon^2} \rceil$$

6 Summary

By corollary 6 in section 5 we know that every finite hypothesis class enjoys UC property. Furthermore, uniform convergence suffices for agnostic PAC learnability. Therefore, every finite hypothesis class is agnostic PAC learnable. That is, if we run the algorithm A with sample $m \geq m_H^{UC}(\epsilon/2, \delta)$, we get

$$\Pr_{S \sim D^m}[L_D(h_S) \leq min_{h' \in H} L_D(h') + \epsilon] \geq 1 - \delta$$

This PAC learning model can be applied to not only binary classification problem, but also multi-class problem and even regression problem. That is, any finite hypothesis H is agnostic PAC learnable with general loss functions, as long as the range loss function is bounded.