

1 The probably approximately correct (PAC) and other learning models[?]

Definition 1.1. Give some basic definition in ML task.

- *Instance space:* domain of X , where X is the sample population. If focus on object with n boolean attributes that is one example with n attributes x_{i1}, \dots, x_{in} , instance space is $\{0, 1\}^n$.
- *Instance:* instance space is a large set and instance is one item in it.
- *Concept:* concept is a subset of instance space.
- *Hypothesis:* also a subset of instance space.

As instance space is understood as sample population, a distribution $P(x)$ can be added to it. Given concept c and hypothesis h , the error of h according to c is

$$Error(h) = \int_{h \Delta c} dP(x)$$

where $h \Delta c = (h \setminus c) \cup (c \setminus h)$.

Next define PAC learnability. Assume for any length of attributes n we have concept sets C_n , which is a subset family of instance space. And define Hypothesis family H_n similarly. An algorithm finds $h \in H_n$ is approximate some concept $c \in C_n$.

Given instance space S . First a $c \in C_n$ is given, for any distribution P over S , $0 < \varepsilon, \delta < 1$, select $N(n, \varepsilon, \delta)$ samples D in c based on distribution P . To be specific, an item in D is $(x, \mathbb{1}_c(x))$ which contains the label(whether x in c or not). If with probability at least $1 - \delta$ A returns h with $Error(h) < \varepsilon$, we call A is PAC learnable.

Remark 1.2. S, C_n are given. First select **any** c, P , D relies on P , h relies on A, D , thus randomness comes from D . Means

$$\mathbb{P}_{D \sim P^{N(n, \varepsilon, \delta)}}(D : Error(h) = Error(A(D)) < \varepsilon) > 1 - \delta, \forall c \in C_n, \forall P$$

Remark 1.3. When $H_n = C_n$, we call it properly PAC learnable. However, most time we do not care what H_n is, any existence of A, H_n we can call C_n PAC learnable.

Remark 1.4. Hypothesis and concept are both subset of instance space, thus can be considered as function $S \rightarrow \{0, 1\}$, where maps to 1 means in the subset.

Given some related results.

- k-DNF concepts[?]: for n boolean variables x_1, \dots, x_n , a k-DNF function has following structure:

$$T_1 \vee \dots \vee T_r$$

where T_i is conjunction of k out of n variables x_1, \dots, x_n . Each concept in k-DNF corresponds to a k-DNF function. Thus algorithm tries to identify the k-DNF function based on given samples. Class of k-DNF is properly learnable for fixed k .

- Perceptrons are PAC learnable[?].

As the size of the hypothesis space increases, it may become easier to find a consistent hypothesis, but it will require more random training examples to insure that this hypothesis is accurate with high probability. **VC-dimension** is an important tool to quantify the complexity of a Hypothesis space[?] [?] [?] [?]. Give its definition

Definition 1.5. Given a hypothesis space H , its VC-dimension is defined as d , which is the maximum number of instances in S , that can be labeled as 1, 0 in all 2^d possible ways. To be specific, assume instances to be x_1, \dots, x_d , then exists $h_1, \dots, h_{2^d} \in H$, subject to $\{(h_i(x_1), \dots, h_i(x_d)) : 1 \leq i \leq 2^d\} = \{1, 0\}^d$.

Criticism about PAC includes concerning the worst case. To be specific, the sample complexity is defined at the worst case over all $c \in C_n$ and all distribution P .

Remark 1.6. Sample complexity refers to $N(n, \epsilon, \delta)$.

If prior information added to possible target concepts or instance space, a Bayesian approach can be used.

Another variant of the PAC model designed to address these issues is the probability of mistake model. Rather than measures the sample complexity, it measures the probability of wrong guess on t -th sample, given previous $t - 1$ training samples.