Two Effective Concept Classes of PACi Incomparable Degrees

Gihanee Senadheera

gihanee.s@siu.edu

Department of Mathematics
Southern Illinois University, Carbondale

October 29, 2020

PAC Learning Model

- PAC stands for **P**robably **A**pproximately **C**orrect
- It is a Machine learning model.
- It was introduced by Leslie Valiant in 1984

PAC Learning Model (Valiant 1984)

Definition

1. Let X be a set, called the *instance space*.

PAC Learning Model (Valiant 1984)

Definition

1. Let X be a set, called the *instance space*.
2. Let C be a subset of $P(X)$ the power set of X, called a *concept class*.

PAC Learning Model (Valiant 1984)

Definition

1. Let X be a set, called the *instance space*.
2. Let C be a subset of $P(X)$ the power set of X, called a *concept class*.
3. The elements of C are called *concepts*.

PAC Learning Model (Valiant 1984)

Definition

We say that C is *PAC Learnable* if and only if there is an algorithm L such that for every $c \in C$, every $\epsilon, \delta \in (0, \frac{1}{2})$ and every probability distribution D on X, the algorithm L behaves as follows:

PAC Learning Model (Valiant 1984)

Definition

We say that C is *PAC Learnable* if and only if there is an algorithm L such that for every $c \in C$, every $\epsilon, \delta \in (0, \frac{1}{2})$ and every probability distribution D on X, the algorithm L behaves as follows:

On input (ϵ, δ), the algorithm L will ask for some number n of examples, and will be given $\{(x_1, i_1), ..., (x_n, i_n)\}$ where x_k are independently randomly drawn from D and $i_k = \chi_c(x_k)$.

PAC Learning Model (Valiant 1984)

Definition

We say that C is *PAC Learnable* if and only if there is an algorithm L such that for every $c \in C$, every $\epsilon, \delta \in (0, \frac{1}{2})$ and every probability distribution D on X, the algorithm L behaves as follows:

On input (ϵ, δ), the algorithm L will ask for some number n of examples, and will be given $\{(x_1, i_1), ..., (x_n, i_n)\}$ where x_k are independently randomly drawn from D and $i_k = \chi_c(x_k)$.

The algorithm will then output some $h \in C$ so that with probability at least $1 - \delta$ in D, the symmetric difference of h and c has the probability at most ϵ in D.

Example

Suppose X is the real line.

- Let C be the set of positive half lines then C is PAC learnable.

Examples

Example

Suppose X is the real line.

- Let C be the set of positive half lines then C is PAC learnable.
- Let C be the set of negative half lines then C is PAC learnable.
- Let C be the set of intervals then C is PAC learnable.

Examples

Example

Suppose X is $\mathbb{R}^2$.

- Let C be the set of axis aligned rectangles then C is PAC learnable.

Examples

Example

Suppose X is $\mathbb{R}^2$.

- Let C be the set of axis aligned rectangles then C is PAC learnable.
- Let C be the set of convex d-gons then C is PAC learnable for any d.

Examples

Example

Suppose $X = \mathbb{R}^d$. Let C be the set of linear-half spaces. Then C is PAC learnable.

Why Π_1^0 classes

A binary tree could explain an interval. For example consider the unit interval.

Why Π_1^0 classes

A binary tree could explain an interval. For example consider the unit interval.

An interval can be seen as a set of paths through a Π_1^0 tree.

Π^0_1 tree

Definition

A relation is Π^0_1 if it is expressed in the form $\forall\ y,\ R(x,y)$ where $R(x,y)$ is computable. The Π^0_1 relations are the co-c.e (complement is c.e) relations. Then Π^0_1 is the set consisting of elements of the form $\{x : \forall y \quad R(x,y)\}$.

Π^0_1 tree

Definition

A relation is Π^0_1 if it is expressed in the form $\forall\ y,\ R(x,y)$ where $R(x,y)$ is computable. The Π^0_1 relations are the co-c.e (complement is c.e) relations. Then Π^0_1 is the set consisting of elements of the form $\{x : \forall y \ \ R(x,y)\}$.

A Π^0_1 tree $T_{e,n}$ is a relation where predecessor relation is a Π^0_1 relation.

Weakly effective concept class

Definition

A weakly effective concept class is a computable enumeration $\varphi_e : \mathbb{N} \to \mathbb{N}$ such that $\varphi_e(n)$ is a Π^0_1 index for a Π^0_1 tree $T_{e,n}$.

An effective concept class

Definition

An effective concept class is a weakly effective concept class $\varphi_e(n)$ such that for each n, the set c_n of paths through $T_{e,n}$ is computable in the sense that there is a computable function $f_{c_n}(d, r) : 2^{<\omega} \times \mathbb{Q} \to \{0, 1\}$ such that

$$f_{c_n}(\sigma, r) = \begin{cases} 1 & \text{if } B_r(\sigma) \cap c_n \neq \varnothing \\ 0 & \text{if } B_{2r}(\sigma) \cap c_n = \varnothing \\ 0 \textit{ or } 1 & \text{otherwise} \end{cases}$$

where $B_r(\sigma)$ is the set of all paths that either extend σ or first differ from it at the $-\lceil lg(r) \rceil$ place or later.

We can say that an effective concept class is a set of Π^0_1 classes. A Π^0_1 class is expressed as the set of infinite paths through a computable tree or the set of infinite paths through a Π^0_1 tree.

Example

Example

The class C of linear half-spaces in $\mathbb{R}^d$ bounded by hyper-planes with computable coefficients is an effective concept class.

Example

Example

The class C of linear half-spaces in $\mathbb{R}^d$ bounded by hyper-planes with computable coefficients is an effective concept class.

Since the distance of a point from the boundary can be computed, the linear half-spaces with computable coefficients is a computable set.

Consider $\mathbb{R}^2$. There are algorithms to compute the distance from a point to a line. The line has computable coefficients. Here no need to use the full precision reals.

Example

Example

The class C of convex d-gons in $\mathbb{R}^2$ with computable vertices is an effective concept class.

PACi Reducibility (Calvert 2018)

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PACi reduces to C', which we denote by $C \leq_{PACi} C'$ exactly when there are functions $g : X \to X'$ and $h : C \to C'$ such that

PACi Reducibility (Calvert 2018)

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PACi reduces to C', which we denote by $C \leq_{PACi} C'$ exactly when there are functions $g : X \to X'$ and $h : C \to C'$ such that

1. g is a Turing functional

PACi Reducibility (Calvert 2018)

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PACi reduces to C', which we denote by $C \leq_{PACi} C'$ exactly when there are functions $g : X \to X'$ and $h : C \to C'$ such that

1. g is a Turing functional
2. h is a computable function on indices

PACi Reducibility (Calvert 2018)

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PACi reduces to C', which we denote by $C \leq_{PACi} C'$ exactly when there are functions $g : X \to X'$ and $h : C \to C'$ such that

1. g is a Turing functional
2. h is a computable function on indices
3. for all $x \in X$ and for all $c \in C$, we have $x \in c$ if and only if $g(x) \in h(c)$.

PACi Reducibility (Calvert 2018)

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PACi reduces to C', which we denote by $C \leq_{PACi} C'$ exactly when there are functions $g : X \to X'$ and $h : C \to C'$ such that

1. g is a Turing functional
2. h is a computable function on indices
3. for all $x \in X$ and for all $c \in C$, we have $x \in c$ if and only if $g(x) \in h(c)$.

The "i" indicates the independence of this definition from size and computation time.

Theorem

Let C and C' be concept classes. Then if C PACi-reduces to C', and C' is PACi learnable, C is PACi learnable.

Theorem

Let C and C' be concept classes. Then if C PACi-reduces to C', and C' is PACi learnable, C is PACi learnable.

Proof:

Let L' be the learning algorithm for C'.
We use L' to learn C.

Theorem

Let C and C' be concept classes. Then if C PACi-reduces to C', and C' is PACi learnable, C is PACi learnable.

Proof:

Let L' be the learning algorithm for C'.
We use L' to learn C.
For a random example (x, c) of the unknown target concept $c \in C$, we can compute the labeled example $(g(x), h(c))$ and give it to L'.

Theorem

Let C and C' be concept classes. Then if C PACi-reduces to C', and C' is PACi learnable, C is PACi learnable.

Proof:

Let L' be the learning algorithm for C'.
We use L' to learn C.
For a random example (x, c) of the unknown target concept $c \in C$, we can compute the labeled example $(g(x), h(c))$ and give it to L'.
If the instance $x \in X$ are drawn according D, then the instances $g(x) \in X'$ are drawn according to some induced distribution D'.

Proof cont.

Although we do not know the target concept c, our definition of reduction guarantees that the computed examples $(g(x), h(c))$ are consistent with some $c' \in C'$, and thus L' will output a hypothesis t' that has error at most ϵ with respect to D'.

Proof cont.

Although we do not know the target concept c, our definition of reduction guarantees that the computed examples $(g(x), h(c))$ are consistent with some $c' \in C'$, and thus L' will output a hypothesis t' that has error at most ϵ with respect to D'.

Our hypothesis for c becomes $t(x) = t'(g(x))$, which has at most ϵ error with respect to D.

PAC Reducibility (Pit, Warmuth '90, Vazirani,Kearns '94

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PAC reduces to C', denoted $C \leq_{PAC} C'$ exactly when $C \leq_{PACi} C'$ via functions g and h such that

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PAC reduces to C', denoted $C \leq_{PAC} C'$ exactly when $C \leq_{PACi} C'$ via functions g and h such that

1. g is computable in polynomial time,

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PAC reduces to C', denoted $C \leq_{PAC} C'$ exactly when $C \leq_{PACi} C'$ via functions g and h such that

1. g is computable in polynomial time,
2. There is a polynomial p such that for any $x \in X$ of size n, the element $g(x)$ is of size at most $p(n)$, and

PAC Reducibility (Pit, Warmuth '90, Vazirani,Kearns '94

Definition

Let C be an effective concept class over the instance space X and C' an effective concept class over the instance space X'.

We say that C PAC reduces to C', denoted $C \leq_{PAC} C'$ exactly when $C \leq_{PACi} C'$ via functions g and h such that

1. g is computable in polynomial time,
2. There is a polynomial p such that for any $x \in X$ of size n, the element $g(x)$ is of size at most $p(n)$, and
3. There is a polynomial q such that for every $c \in C$ of size n, the concept $h(c)$ is of size at most $q(n)$.

- Observe that empty concept class on the empty instance space is reducible to any other concept class.

- Observe that empty concept class on the empty instance space is reducible to any other concept class.
- Also any concept class is reducible to itself through the identity function.

- Observe that empty concept class on the empty instance space is reducible to any other concept class.
- Also any concept class is reducible to itself through the identity function.
- We can infer that there are $\leq_{PAC}$ incomparable concept classes since there are continuum many concept classes on a countably infinite instance spaces.

- Observe that empty concept class on the empty instance space is reducible to any other concept class.
- Also any concept class is reducible to itself through the identity function.
- We can infer that there are $\leq_{PAC}$ incomparable concept classes since there are continuum many concept classes on a countably infinite instance spaces.
- This degree structure is analogous to Turing degrees and their structures. So, we can expect the effective concept classes to behave in similar manner to computably enumerable degrees.

PACi Degree

Definition

We say $C \sim C'$ if $C \leq_{PAC_i} C'$ and $C' \leq_{PAC_i} C$, the relation $\sim$ is an equivalence class. The PACi degree of concept class C is $deg(C) = \{C' : C' \sim C\}$

Examples

Example

Let X be the empty instance space and X' be any instance space.

Let C the empty concept class over X and C' be any concept class over X'.

Examples

Example

Let X be the empty instance space and X' be any instance space.

Let C the empty concept class over X and C' be any concept class over X'.

Define $g : X \to X'$ Turing functional and $h : C \to C'$ a computable functional on indices.

Examples

Example

Let X be the empty instance space and X' be any instance space.

Let C the empty concept class over X and C' be any concept class over X'.

Define $g : X \to X'$ Turing functional and $h : C \to C'$ a computable functional on indices.

Then *for all* $x \in X$ and *for all* $c \in C$ we have $x \in c$ iff $g(x) \in h(c)$. We can write $C \leq_{PACi} C'$.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

The positive half lines are bounded below. If positive half lines are bounded below by computable lower bound then the concept class C is an effective concept class.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

The positive half lines are bounded below. If positive half lines are bounded below by computable lower bound then the concept class C is an effective concept class.

Similarly we can show that C' is also an effective concept class.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

The positive half lines are bounded below. If positive half lines are bounded below by computable lower bound then the concept class C is an effective concept class.

Similarly we can show that C' is also an effective concept class.

Define $g : \mathbb{R} \to \mathbb{R}$ by $g(x) = -x$ and $h : C \to C'$ by $h((a, \infty)) = (-\infty, -a)$.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

The positive half lines are bounded below. If positive half lines are bounded below by computable lower bound then the concept class C is an effective concept class.

Similarly we can show that C' is also an effective concept class.

Define $g : \mathbb{R} \to \mathbb{R}$ by $g(x) = -x$ and $h : C \to C'$ by $h((a, \infty)) = (-\infty, -a)$.

Now we can show that *for all* $x \in \mathbb{R}$ and *for all* positive half lines $c = (a, \infty)$ in C we have $x \in c$ iff $g(x) \in h(c)$ where $h(c)$ is a negative half line.

Examples

Example

Let $X = X' = \mathbb{R}$ be the two instance spaces.
Let C be the set of positive half lines and C' be the set of negative half lines.

The positive half lines are bounded below. If positive half lines are bounded below by computable lower bound then the concept class C is an effective concept class.

Similarly we can show that C' is also an effective concept class.

Define $g : \mathbb{R} \to \mathbb{R}$ by $g(x) = -x$ and $h : C \to C'$ by $h((a, \infty)) = (-\infty, -a)$.

Now we can show that *for all* $x \in \mathbb{R}$ and *for all* positive half lines $c = (a, \infty)$ in C we have $x \in c$ iff $g(x) \in h(c)$ where $h(c)$ is a negative half line.

This will give us $C \leq_{PACi} C'$. With appropriate functionals we can show that $C' \leq_{PACi} C$. Thus $C \sim C'$.

Friedberg Muchnik Theorem [1957, 1956]

Theorem

There exist computably enumerable (c.e.) sets A and B such that $A \not\leq_T B$ and $B \not\leq_T A$.

Friedberg Muchnik Theorem [1957, 1956]

Theorem

There exist computably enumerable (c.e.) sets A and B such that $A \nleq_T B$ and $B \nleq_T A$.

Idea of the Proof:
It suffice to recursively enumerate A and B to meet for all e the requirements:

$$R_{2e} : A \neq \{e\}^B$$
$$R_{2e+1} : B \neq \{e\}^A$$

The strategy for meeting a single such requirement R_{2e} is to attach to R_{2e} a potential "witness" x not yet enumerated in A and to look for a stage $s+1$ such that

$$\{e\}_s^{B_s}(x)\downarrow = 0.$$

The strategy for meeting a single such requirement R_{2e} is to attach to R_{2e} a potential "witness" x not yet enumerated in A and to look for a stage $s+1$ such that

$$\{e\}_s^{B_s}(x)\downarrow = 0.$$

If no such stage exists we do nothing and R_{2e} is automatically satisfied by the witness x because $A(x) = 0$ and either $\{e\}^B(x)\uparrow$ or $\{e\}^B(x)\downarrow \neq 0$.

The strategy for meeting a single such requirement R_{2e} is to attach to R_{2e} a potential "witness" x not yet enumerated in A and to look for a stage $s+1$ such that

$$\{e\}_s^{B_s}(x) \downarrow = 0.$$

If no such stage exists we do nothing and R_{2e} is automatically satisfied by the witness x because $A(x) = 0$ and either $\{e\}^B(x) \uparrow$ or $\{e\}^B(x) \downarrow \neq 0$.
If $s+1$ exists, we say R_{2e} *requires attention* at stage $s+1$. Now R_{2e} *receives attention* and we: (1) enumerate x in A_{s+1};

The strategy for meeting a single such requirement R_{2e} is to attach to R_{2e} a potential "witness" x not yet enumerated in A and to look for a stage $s+1$ such that

$$\{e\}_s^{B_s}(x) \downarrow = 0.$$

If no such stage exists we do nothing and R_{2e} is automatically satisfied by the witness x because $A(x) = 0$ and either $\{e\}^B(x) \uparrow$ or $\{e\}^B(x) \downarrow \neq 0$.
If $s+1$ exists, we say R_{2e} *requires attention* at stage $s+1$. Now R_{2e} *receives attention* and we: (1) enumerate x in A_{s+1}; (2) define the *restraint function* $r(2e, s+1)$ and attempt (with priority R_{2e}) to restrain any number $y \leq r = r(2e, s+1)$ from later entering B. If we achieve the latter objective then

$$\{e\}^B(x) = 0.$$

The strategy for meeting a single such requirement R_{2e} is to attach to R_{2e} a potential "witness" x not yet enumerated in A and to look for a stage $s+1$ such that

$$\{e\}_s^{B_s}(x)\downarrow = 0.$$

If no such stage exists we do nothing and R_{2e} is automatically satisfied by the witness x because $A(x)=0$ and either $\{e\}^B(x)\uparrow$ or $\{e\}^B(x)\downarrow\neq 0$.

If $s+1$ exists, we say R_{2e} *requires attention* at stage $s+1$. Now R_{2e} *receives attention* and we: (1) enumerate x in A_{s+1}; (2) define the *restraint function* $r(2e,s+1)$ and attempt (with priority R_{2e}) to restrain any number $y\leq r=r(2e,s+1)$ from later entering B. If we achieve the latter objective then

$$\{e\}^B(x)=0.$$

However, $A(x)=1$ so requirement R_{2e} is satisfied. (The strategy for R_{2e+1} is the same but with the roles of A and B reversed.)

Theorem

There exists an effective concept class C over the instance space $X = 2^\omega$ and an effective concept class C' over the instance space $X' = 2^\omega$ such that C does not PACi reduce to C' and also C' does not PACi reduce to C (i.e. $C \not\leq_{PACi} C'$ and $C' \not\leq_{PACi} C$).

Proof:

The two concept classes C and C' are constructed over the instance spaces X and X' respectively. Let $\{h_t | t \in \mathbb{N}\}$ enumerate the set of all computable functions from $\mathbb{N} \to \mathbb{N}$.

Requirements : R_{2t} : there exists $c \in C$ such that $h_t(c) \notin C'$

R_{2t+1} : there exists $c' \in C'$ such that $h_t(c') \notin C$

Satisfying one requirement

Consider R_{2t}.

To satisfy the requirement R_{2t} we will attach a potential witness c: a concept, to R_{2t} which is not yet enumerated in C.

We choose c such that c is an index for a tree.

Satisfying one requirement cont.

At stage s pick a c such that $c \notin B_s$ and $c \notin C_s$ and $h_t(c) \notin C'_s$.

Let B_s be the set of all trees that can not be enumerated in C_s.

We will enumerate c in C_s and enumerate $h_t(c)$ in A_s.

Let A_s be the set of all trees that can not be enumerated in C'.

Thus we restrain the tree $h_t(c)$ later entering to C'.

This is achieved by checking the condition, $c' \notin A_{s+1}$.

Satisfying one requirement cont.

Since $C_s \not\leq_{PACi} C'_s$ we have $C \not\leq_{PACi} C'$.

The strategy for R_{2t+1} is the same but with roles of C_s and C'_s reversed.

We call the sets A and B as restraint sets.

Construction of the two concept classes, C and C'.

Let $X = X' = 2^{\omega}$.

Let $\{c_n\}_{n=1}^{\infty}$ be a family of trees, where c_n has n number of 1's and followed by zeros. In this sequence each of these trees c_n, consists of a single infinite path.

Construction of the two concept classes, C and C'.

Let $X = X' = 2^{\omega}$.

Let $\{c_n\}_{n=1}^{\infty}$ be a family of trees, where c_n has n number of 1's and followed by zeros. In this sequence each of these trees c_n, consists of a single infinite path.

Stage $s = 0$: Let $C_0 = C'_0 = \phi$ and $A_0 = B_0 = \phi$.

Stage $s + 1$:

Requirement R_{2t} requires attention if, we have not enumerated a witness, $c \in C$ for the requirement R_{2t}.

Requirement R_{2t+1} requires attention if, we have not enumerated a witness, $c' \in C'$ for the requirement R_{2t+1}.

Construction of the two concept classes, C and C'. Cont.

Chose least $i \leq s$ such that R_i requires attention.

Suppose $i = 2t$. Now R_{2t} receives attention. Pick a tree c from the family $\{c_n\}$ defined above such that $c \notin C_s$ and $c \notin B_s$ and $h_t(c) \notin C'_s$. Enumerate $c \in C_{s+1}$ and $h_t(c)$ in A_{s+1}.

Construction of the two concept classes, C and C'. Cont.

Chose least $i \leq s$ such that R_i requires attention.

Suppose $i = 2t$. Now R_{2t} receives attention. Pick a tree c from the family $\{c_n\}$ defined above such that $c \notin C_s$ and $c \notin B_s$ and $h_t(c) \notin C'_s$. Enumerate $c \in C_{s+1}$ and $h_t(c)$ in A_{s+1}.

Suppose $i = 2t + 1$. Now R_{2t+1} receives attention. Pick a tree c' from the family $\{c_n\}$ such that $c' \notin C'_s$ and $c' \notin A_s$ and $h_t(c') \notin C_s$. Then enumerate $c' \in C'_{s+1}$ and $h_t(c')$ in B_{s+1}.

At each stage we will be checking through finite amount of trees in C_s, C'_s, A_s or B_s.

When a requirement is satisfied at stage s it will remain satisfied forever.

Future Work

To show there exists two effective concept classes of PAC incomparable degree

Thank you!