The intrinsic complexity of algorithmic learning

A logical and combinatorial perspective

John Goodrick

Universidad de los Andes
Bogotá, Colombia

Escuela de Física-Matemática
30 de mayo, 2019

First example

Example:[1] Suppose the ripeness of a *lulo* is a function of its firmness and color.

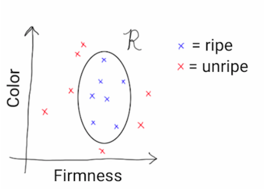


Image credit: "Fibonacci," CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=564934

We can learn the concept of "ripeness" from a small set of labeled examples, if we know that the region $\mathscr{R}$ is of a simple geometric form (e.g. the interior of a rectangle or ellipse)...

...whereas if the class of possible $\mathscr{R}$ is very complicated (e.g. with many fractal-like sets), maybe this learning task is impossible.

[1] Adapted from Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning*

First example

Example:[1] Suppose the ripeness of a *lulo* is a function of its firmness and color.

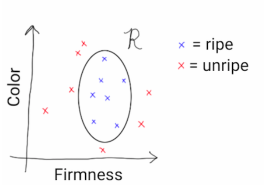


Image credit: "Fibonacci," CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=564934

We can learn the concept of "ripeness" from a small set of labeled examples, if we know that the region $\mathscr{R}$ is of a simple geometric form (e.g. the interior of a rectangle or ellipse)...

...whereas if the class of possible $\mathscr{R}$ is very complicated (e.g. with many fractal-like sets), maybe this learning task is impossible.

[1] Adapted from Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning*

First example

Example:[1] Suppose the ripeness of a *lulo* is a function of its firmness and color.

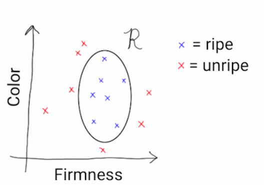


Image credit: "Fibonacci," CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=564934

We can learn the concept of "ripeness" from a small set of labeled examples, if we know that the region $\mathscr{R}$ is of a simple geometric form (e.g. the interior of a rectangle or ellipse)...

...whereas if the class of possible $\mathscr{R}$ is very complicated (e.g. with many fractal-like sets), maybe this learning task is impossible.

---

[1] Adapted from Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning*

How should we model algorithmic learning?

Questions: How should we model the concept of a “learning task” mathematically?

Which learning tasks are inherently easy, difficult, or impossible?

Algorithmic learning theory attempts to answer these questions, just as the study of Turing machines attempts to define what is, in principle, computable.

Some goals of the theory:

- Find elegant ways to characterize learnability of concepts;
- Find bounds on the number of samples needed to learn concepts.

How should we model algorithmic learning?

Questions: How should we model the concept of a "learning task" mathematically?

Which learning tasks are inherently easy, difficult, or impossible?

Algorithmic learning theory attempts to answer these questions, just as the study of Turing machines attempts to define what is, in principle, computable.

Some goals of the theory:

- Find elegant ways to characterize learnability of concepts;
- Find bounds on the number of samples needed to learn concepts.

How should we model algorithmic learning?

Questions: How should we model the concept of a "learning task" mathematically?

Which learning tasks are inherently easy, difficult, or impossible?

Algorithmic learning theory attempts to answer these questions, just as the study of Turing machines attempts to define what is, in principle, computable.

Some goals of the theory:

- Find elegant ways to characterize learnability of concepts;
- Find bounds on the number of samples needed to learn concepts.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

Outline of Talk

1. PAC learning ("Probably Approximately Correct")
2. Vapnik-Chervonenkis dimension
3. VC bounds for perceptrons and neural nets
4. Other learning models (online learning, etc.)
5. Current directions

DISCLAIMER: This is an expository talk; none of the results presented here are original.

PAC learning: introduction

An easy learning task: A Martian wants to learn which range Earthlings call "room temperature." She has n labeled samples

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

[$y_i = 1$ if x_i degrees C is room temperature, $y_i = 0$ otherwise].

She might guess the interval $[x_{i_0}, x_{i_1}]$ bounded by the minimum x_{i_0} and maximum x_{i_1} from S which are labeled by 1.

This is a good strategy, *even if we do not know the distribution by which S was selected.*

PAC learning: introduction

An easy learning task: A Martian wants to learn which range Earthlings call "room temperature." She has n labeled samples

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

[$y_i = 1$ if x_i degrees C is room temperature, $y_i = 0$ otherwise].

She might guess the interval $[x_{i_0}, x_{i_1}]$ bounded by the minimum x_{i_0} and maximum x_{i_1} from S which are labeled by 1.

This is a good strategy, *even if we do not know the distribution by which S was selected.*

PAC learning: introduction

An easy learning task: A Martian wants to learn which range Earthlings call "room temperature." She has n labeled samples

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

[$y_i = 1$ if x_i degrees C is room temperature, $y_i = 0$ otherwise].

She might guess the interval $[x_{i_0}, x_{i_1}]$ bounded by the minimum x_{i_0} and maximum x_{i_1} from S which are labeled by 1.

This is a good strategy, *even if we do not know the distribution by which S was selected.*

Concept classes

- X is a set of *instances* (data points we wish to classify);
- A *concept* is any $C \subseteq X$, equivalently $\chi_C : X \to \{0,1\}$;
- A *sample (labeled by C)* is a finite multiset

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

  with $x_i \in X$, $y_i \in \{0,1\}$ (and $y_i = 1$ iff $x_i \in C$);
- A *learning algorithm* is **any** function

$$A : (X \times \{0,1\})^{<\omega} \to \mathscr{H}$$

  from the set of all possible samples into a set $\mathscr{H} \subseteq \mathscr{P}(X)$ of possible *hypotheses*. (And usually we assume $C \in \mathscr{H}$.)

WARNING: In the case where $\mathscr{H}$ is uncountable, we should make some extra measurability assumptions. In particular, we could assume X is a standard Borel space, $\mathscr{H} = \{h_t : t \in [0,1]\}$ is parameterized by $t \in [0,1]$, and $\{(x,t) : x \in h_t\}$ is the image of a Borel set under a continuous map.

A loss function

Generally we will consider a probability distribution μ on the instances X, and consider samples

$$S = ((x_1, y_1), \ldots, (x_n, y_n))$$

labeled by some $C \in \mathscr{H}$, with x_i selected independently according to μ (that is, $S \sim \mu^n$).

The *loss function* applied to a hypothesis $h \subseteq X$ is

$$L_{\mu,C}(h) = \Pr_{x \sim \mu}\left[(C \setminus h) \cup (h \setminus C)\right],$$

i.e. the probability that an x selected randomly by μ is misclassified by h.

A loss function

Generally we will consider a probability distribution μ on the instances X, and consider samples

$$S = ((x_1, y_1), \ldots, (x_n, y_n))$$

labeled by some $C \in \mathscr{H}$, with x_i selected independently according to μ (that is, $S \sim \mu^n$).

The *loss function* applied to a hypothesis $h \subseteq X$ is

$$L_{\mu,C}(h) = \Pr_{x \sim \mu} \left[(C \setminus h) \cup (h \setminus C) \right],$$

i.e. the probability that an x selected randomly by μ is misclassified by h.

PAC learning: the definition (Valiant 1984)

The concept class $\mathscr{H} \subseteq \mathscr{P}(X)$ is *PAC learnable* ("Probably Approximately Correct") if there are

- a learning algorithm $A : (X \times \{0,1\})^{<\omega} \to \mathscr{H}$, and
- a function $m : (0,1)^2 \to \mathbb{N}$,

such that

- for **any** $C \in \mathscr{H}$, **any** $\delta, \epsilon \in (0,1)$, and **any** probability distribution μ on X,
- and for any n "big enough" (that is, $n \geq m(\delta, \epsilon)$),

$$\Pr_{S \sim \mu^n} [L_{\mu,C}(A(S)) \leq \epsilon] \geq 1 - \delta.$$

Recall that $L_{\mu,C}$ is the loss function.

Note that the bound $m(\delta, \epsilon)$ does not depend on μ nor on C!

PAC learning: the definition (Valiant 1984)

The concept class $\mathscr{H} \subseteq \mathscr{P}(X)$ is *PAC learnable* ("Probably Approximately Correct") if there are

- a learning algorithm $A : (X \times \{0,1\})^{<\omega} \to \mathscr{H}$, and
- a function $m : (0,1)^2 \to \mathbb{N}$,

such that

- for **any** $C \in \mathscr{H}$, **any** $\delta, \epsilon \in (0,1)$, and **any** probability distribution μ on X,
- and for any n "big enough" (that is, $n \geq m(\delta, \epsilon)$),

$$\Pr_{S \sim \mu^n} [L_{\mu,C}(A(S)) \leq \epsilon] \geq 1 - \delta.$$

Recall that $L_{\mu,C}$ is the loss function.

Note that the bound $m(\delta, \epsilon)$ does not depend on μ nor on C!

PAC learning: the definition (Valiant 1984)

The concept class $\mathscr{H} \subseteq \mathscr{P}(X)$ is *PAC learnable* ("Probably Approximately Correct") if there are

- a learning algorithm $A : (X \times \{0,1\})^{<\omega} \to \mathscr{H}$, and
- a function $m : (0,1)^2 \to \mathbb{N}$,

such that

- for **any** $C \in \mathscr{H}$, **any** $\delta, \epsilon \in (0,1)$, and **any** probability distribution μ on X,
- and for any n "big enough" (that is, $n \geq m(\delta, \epsilon)$),

$$\mathop{\mathbf{Pr}}_{S \sim \mu^n} [L_{\mu,C}(A(S)) \leq \epsilon] \geq 1 - \delta.$$

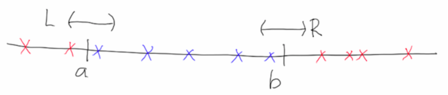
Recall that $L_{\mu,C}$ is the loss function.

Note that the bound $m(\delta, \epsilon)$ does not depend on μ nor on C!

PAC learning: a simple example

If $X = \mathbb{R}$ and $\mathscr{H} = \{[a, b] : a, b \in \mathbb{R}\}$ is the class of all closed bounded intervals, then $\mathscr{H}$ is PAC learnable, with bound $m(\delta, \epsilon) = \frac{2}{\epsilon} \ln(\frac{2}{\delta})$.

Proof:



Say $A(S)$ selects an interval consistent with S. Pick intervals L and R containing a and b respectively such that $\mu(L) = \mu(R) = \frac{\epsilon}{2}$. Then

$$\Pr_{S \sim \mu^n} [S \cap L = \emptyset, S \cap R = \emptyset] \leq 2 \cdot (1 - \frac{\epsilon}{2})^n \leq 2e^{-\frac{\epsilon n}{2}},$$

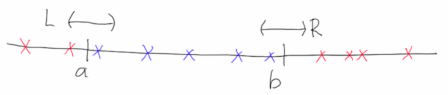
so if $n \geq \frac{2}{\epsilon} \ln(\frac{2}{\delta})$, then S will contain points from both L and R with probability at least $1 - \delta$. But if S contains instances of both L and R then $L_{\mu, C}(A(S)) \leq \epsilon$.

Proof adapted from Example 2.3.1 of *NIP Theories and Computational Learning Theory* by Vincent Guingona.

PAC learning: a simple example

If $X = \mathbb{R}$ and $\mathscr{H} = \{[a, b] : a, b \in \mathbb{R}\}$ is the class of all closed bounded intervals, then $\mathscr{H}$ is PAC learnable, with bound $m(\delta, \epsilon) = \frac{2}{\epsilon} \ln(\frac{2}{\delta})$.

Proof:



Say $A(S)$ selects an interval consistent with S. Pick intervals L and R containing a and b respectively such that $\mu(L) = \mu(R) = \frac{\epsilon}{2}$. Then

$$\Pr_{S \sim \mu^n} [S \cap L = \emptyset, S \cap R = \emptyset] \leq 2 \cdot (1 - \frac{\epsilon}{2})^n \leq 2e^{-\frac{\epsilon n}{2}},$$

so if $n \geq \frac{2}{\epsilon} \ln(\frac{2}{\delta})$, then S will contain points from both L and R with probability at least $1 - \delta$. But if S contains instances of both L and R then $L_{\mu, C}(A(S)) \leq \epsilon$.

Proof adapted from Example 2.3.1 of *NIP Theories and Computational Learning Theory* by Vincent Guingona.

Other examples?

That was way too tricky!

Question: Is there an easier way to determine whether simple classes are PAC learnable without $\delta - \epsilon$ manipulations?

Answer: YES, with Vapnik-Chervonenkis dimension.

Other examples?

That was way too tricky!

Question: Is there an easier way to determine whether simple classes are PAC learnable without $\delta - \epsilon$ manipulations?

Answer: YES, with Vapnik-Chervonenkis dimension.

Other examples?

That was way too tricky!

Question: Is there an easier way to determine whether simple classes are PAC learnable without $\delta - \epsilon$ manipulations?

Answer: YES, with Vapnik-Chervonenkis dimension.

Vapnik-Chervonenkis dimension

Say $\mathscr{H} \subseteq \mathscr{P}(X)$ is a concept class (set of subsets of X).

1. If $A \subseteq X$, then $\mathscr{H}$ *shatters the set* A if for every $B \subseteq A$ there is some $h_B \in \mathscr{H}$ such that $h_B \cap A = B$.
2. The *Vapnik-Chervonenkis dimension* of $\mathscr{H}$ is

$$\mathrm{VCdim}(\mathscr{H}) = \max\{\|A\| : A \subseteq X \text{ and } \mathscr{H} \text{ shatters } A\},$$

   the maximum size of a subset of X shattered by $\mathscr{H}$ ($\mathrm{VCdim}(\mathscr{H}) = \infty$ if there is no such finite bound).

The class $\mathscr{H} = \{h_1, h_2, h_3, h_4\}$ shatters A. $\mathrm{VCdim}(\mathscr{H}) = 2$.

Easy bound: If $\mathscr{H}$ is finite, then $\mathrm{VCdim}(\mathscr{H}) \leq \log_2(\|\mathscr{H}\|)$.

Vapnik-Chervonenkis dimension

Say $\mathscr{H} \subseteq \mathscr{P}(X)$ is a concept class (set of subsets of X).

1. If $A \subseteq X$, then $\mathscr{H}$ *shatters the set* A if for every $B \subseteq A$ there is some $h_B \in \mathscr{H}$ such that $h_B \cap A = B$.
2. The *Vapnik-Chervonenkis dimension* of $\mathscr{H}$ is

$$\mathrm{VCdim}(\mathscr{H}) = \max\{\|A\| : A \subseteq X \text{ and } \mathscr{H} \text{ shatters } A\},$$

   the maximum size of a subset of X shattered by $\mathscr{H}$ ($\mathrm{VCdim}(\mathscr{H}) = \infty$ if there is no such finite bound).

The class $\mathscr{H} = \{h_1, h_2, h_3, h_4\}$ shatters A. $\mathrm{VCdim}(\mathscr{H}) = 2$.

Easy bound: If $\mathscr{H}$ is finite, then $\mathrm{VCdim}(\mathscr{H}) \leq \log_2(\|\mathscr{H}\|)$.

Vapnik-Chervonenkis dimension

Say $\mathscr{H} \subseteq \mathscr{P}(X)$ is a concept class (set of subsets of X).

1. If $A \subseteq X$, then $\mathscr{H}$ *shatters the set* A if for every $B \subseteq A$ there is some $h_B \in \mathscr{H}$ such that $h_B \cap A = B$.
2. The *Vapnik-Chervonenkis dimension* of $\mathscr{H}$ is

$$\mathrm{VCdim}(\mathscr{H}) = \max\{\|A\| : A \subseteq X \text{ and } \mathscr{H} \text{ shatters } A\},$$

   the maximum size of a subset of X shattered by $\mathscr{H}$ ($\mathrm{VCdim}(\mathscr{H}) = \infty$ if there is no such finite bound).

The class $\mathscr{H} = \{h_1, h_2, h_3, h_4\}$ shatters A. $\mathrm{VCdim}(\mathscr{H}) = 2$.

Easy bound: If $\mathscr{H}$ is finite, then $\mathrm{VCdim}(\mathscr{H}) \leq \log_2(\|\mathscr{H}\|)$.

Vapnik-Chervonenkis dimension

Say $\mathscr{H} \subseteq \mathscr{P}(X)$ is a concept class (set of subsets of X).

1. If $A \subseteq X$, then $\mathscr{H}$ *shatters the set* A if for every $B \subseteq A$ there is some $h_B \in \mathscr{H}$ such that $h_B \cap A = B$.
2. The *Vapnik-Chervonenkis dimension* of $\mathscr{H}$ is

$$\mathrm{VCdim}(\mathscr{H}) = \max\{\|A\| : A \subseteq X \text{ and } \mathscr{H} \text{ shatters } A\},$$

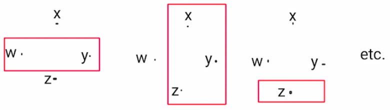
   the maximum size of a subset of X shattered by $\mathscr{H}$ ($\mathrm{VCdim}(\mathscr{H}) = \infty$ if there is no such finite bound).

The class $\mathscr{H} = \{h_1, h_2, h_3, h_4\}$ shatters A. $\mathrm{VCdim}(\mathscr{H}) = 2$.

Easy bound: If $\mathscr{H}$ is finite, then $\mathrm{VCdim}(\mathscr{H}) \leq \log_2(\|\mathscr{H}\|)$.

Simple examples of Vapnik-Chervonenkis dimension

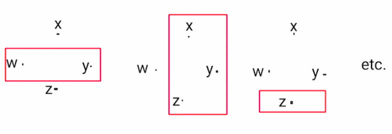
- If $\mathscr{H}_{int}$ is the class of all closed bounded intervals in $\mathbb{R}$, then $\mathrm{VCdim}(\mathscr{H}_{int}) = 2$. (If $x < y < z$, then there is no interval $[a, b]$ such that $[a, b] \cap \{x, y, z\} = \{x, z\}$.)
- If $\mathscr{H}_{box}$ is the class of all closed boxes $[a, b] \times [c, d]$ in $\mathbb{R}^2$, then $\mathrm{VCdim}(\mathscr{H}_{box}) = 4$.



- If $\mathscr{H}_{fin}$ is the set of all finite subsets of $\mathbb{N}$, then $\mathrm{VCdim}(\mathscr{H}_{fin}) = \infty$.
- The set of all interiors of convex polygons in $\mathbb{R}^2$ has infinite VC-dimension. (Exercise!)

Simple examples of Vapnik-Chervonenkis dimension

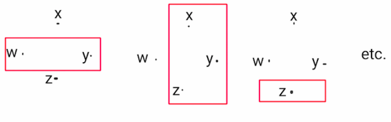
- If $\mathscr{H}_{int}$ is the class of all closed bounded intervals in $\mathbb{R}$, then $\mathrm{VCdim}(\mathscr{H}_{int}) = 2$. (If $x < y < z$, then there is no interval $[a, b]$ such that $[a, b] \cap \{x, y, z\} = \{x, z\}$.)
- If $\mathscr{H}_{box}$ is the class of all closed boxes $[a, b] \times [c, d]$ in $\mathbb{R}^2$, then $\mathrm{VCdim}(\mathscr{H}_{box}) = 4$.



- If $\mathscr{H}_{fin}$ is the set of all finite subsets of $\mathbb{N}$, then $\mathrm{VCdim}(\mathscr{H}_{fin}) = \infty$.
- The set of all interiors of convex polygons in $\mathbb{R}^2$ has infinite VC-dimension. (Exercise!)

Simple examples of Vapnik-Chervonenkis dimension

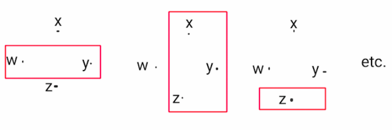
- If $\mathscr{H}_{int}$ is the class of all closed bounded intervals in $\mathbb{R}$, then $\mathrm{VCdim}(\mathscr{H}_{int}) = 2$. (If $x < y < z$, then there is no interval $[a, b]$ such that $[a, b] \cap \{x, y, z\} = \{x, z\}$.)
- If $\mathscr{H}_{box}$ is the class of all closed boxes $[a, b] \times [c, d]$ in $\mathbb{R}^2$, then $\mathrm{VCdim}(\mathscr{H}_{box}) = 4$.



- If $\mathscr{H}_{fin}$ is the set of all finite subsets of $\mathbb{N}$, then $\mathrm{VCdim}(\mathscr{H}_{fin}) = \infty$.
- The set of all interiors of convex polygons in $\mathbb{R}^2$ has infinite VC-dimension. (Exercise!)

Simple examples of Vapnik-Chervonenkis dimension

- If $\mathscr{H}_{int}$ is the class of all closed bounded intervals in $\mathbb{R}$, then $\text{VCdim}(\mathscr{H}_{int}) = 2$. (If $x < y < z$, then there is no interval $[a, b]$ such that $[a, b] \cap \{x, y, z\} = \{x, z\}$.)
- If $\mathscr{H}_{box}$ is the class of all closed boxes $[a, b] \times [c, d]$ in $\mathbb{R}^2$, then $\text{VCdim}(\mathscr{H}_{box}) = 4$.



- If $\mathscr{H}_{fin}$ is the set of all finite subsets of $\mathbb{N}$, then $\text{VCdim}(\mathscr{H}_{fin}) = \infty$.
- The set of all interiors of convex polygons in $\mathbb{R}^2$ has infinite VC-dimension. (Exercise!)

VC dimension and PAC learnability

Theorem (Blumer, Ehrenfeucht, Haussler and Warmuth, '89)
$\mathscr{H}$ is PAC learnable if and only if $\mathrm{VCdim}(\mathscr{H}) < \infty$.

In fact, if $\mathrm{VCdim}(\mathscr{H}) = d$, then $\mathscr{H}$ is PAC learnable with bound

$$m(\delta, \epsilon) = \max\left(\frac{4}{\epsilon}\log_2\left(\frac{2}{\delta}\right), \frac{8d}{\epsilon}\log_2\left(\frac{13}{\epsilon}\right)\right)$$

$$= O\left(d\log\left(1/\delta\right)1/\epsilon\log\left(1/\epsilon\right)\right).$$

In other words, if we want error at most $\leq \epsilon$ with probability at least $1-\delta$, it is sufficient to train with $m(\delta, \epsilon)$ data points.

Thus the class $\mathscr{H}_{fin}$ of all finite subsets of $\mathbb{N}$ is not PAC learnable, nor is the class of all convex polygons in the plane.

VC dimension and PAC learnability

Theorem (Blumer, Ehrenfeucht, Haussler and Warmuth, '89)
$\mathscr{H}$ is PAC learnable if and only if VCdim($\mathscr{H}$) $< \infty$.

In fact, if VCdim($\mathscr{H}$) $= d$, then $\mathscr{H}$ is PAC learnable with bound

$$m(\delta, \epsilon) = \max\left(\frac{4}{\epsilon}\log_2\left(\frac{2}{\delta}\right), \frac{8d}{\epsilon}\log_2\left(\frac{13}{\epsilon}\right)\right)$$

$$= O\left(d \log\left(1/\delta\right) 1/\epsilon \log\left(1/\epsilon\right)\right).$$

In other words, if we want error at most $\leq \epsilon$ with probability at least $1 - \delta$, it is sufficient to train with $m(\delta, \epsilon)$ data points.

Thus the class $\mathscr{H}_{fin}$ of all finite subsets of $\mathbb{N}$ is not PAC learnable, nor is the class of all convex polygons in the plane.

VC dimension and PAC learnability

Theorem (Blumer, Ehrenfeucht, Haussler and Warmuth, '89)
$\mathscr{H}$ is PAC learnable if and only if $\mathrm{VCdim}(\mathscr{H}) < \infty$.

In fact, if $\mathrm{VCdim}(\mathscr{H}) = d$, then $\mathscr{H}$ is PAC learnable with bound

$$m(\delta, \epsilon) = \max\left(\frac{4}{\epsilon}\log_2\left(\frac{2}{\delta}\right), \frac{8d}{\epsilon}\log_2\left(\frac{13}{\epsilon}\right)\right)$$

$$= O\left(d \log\left(1/\delta\right) 1/\epsilon \log\left(1/\epsilon\right)\right).$$

In other words, if we want error at most $\leq \epsilon$ with probability at least $1 - \delta$, it is sufficient to train with $m(\delta, \epsilon)$ data points.

Thus the class $\mathscr{H}_{fin}$ of all finite subsets of $\mathbb{N}$ is not PAC learnable, nor is the class of all convex polygons in the plane.

VC dimension and PAC learnability

Theorem (Blumer, Ehrenfeucht, Haussler and Warmuth, '89)
$\mathscr{H}$ is PAC learnable if and only if $\mathrm{VCdim}(\mathscr{H}) < \infty$.

In fact, if $\mathrm{VCdim}(\mathscr{H}) = d$, then $\mathscr{H}$ is PAC learnable with bound

$$m(\delta, \epsilon) = \max\left(\frac{4}{\epsilon}\log_2\left(\frac{2}{\delta}\right), \frac{8d}{\epsilon}\log_2\left(\frac{13}{\epsilon}\right)\right)$$

$$= O\left(d \log\left(1/\delta\right) 1/\epsilon \log\left(1/\epsilon\right)\right).$$

In other words, if we want error at most $\leq \epsilon$ with probability at least $1 - \delta$, it is sufficient to train with $m(\delta, \epsilon)$ data points.

Thus the class $\mathscr{H}_{fin}$ of all finite subsets of $\mathbb{N}$ is not PAC learnable, nor is the class of all convex polygons in the plane.

VC dimension and growth function

An important tool for studying VC dimension is the **growth function.** Let

$$\mathscr{H}_A = \{C \cap A : C \in \mathscr{H}\}$$

and define $\pi_{\mathscr{H}} : m \to m$ by

$$\pi_{\mathscr{H}}(m) = \max \{\|\mathscr{H}_A\| : A \subseteq X, \|A\| = m\}.$$

Lemma (Sauer-Shelah): For any $\mathscr{H}$, either

- $\mathrm{VCdim}(\mathscr{H}) = d$ and $\pi_{\mathscr{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} = O(m^d)$,
- or else $\mathrm{VCdim}(\mathscr{H}) = \infty$ and $\pi_{\mathscr{H}}(m) = 2^m$.

VC dimension and growth function

An important tool for studying VC dimension is the **growth function.** Let

$$\mathscr{H}_A = \{C \cap A : C \in \mathscr{H}\}$$

and define $\pi_{\mathscr{H}} : m \to m$ by

$$\pi_{\mathscr{H}}(m) = \max\{\|\mathscr{H}_A\| : A \subseteq X, \|A\| = m\}.$$

Lemma (Sauer-Shelah): For any $\mathscr{H}$, either

- $\mathrm{VCdim}(\mathscr{H}) = d$ and $\pi_{\mathscr{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} = O(m^d)$,
- or else $\mathrm{VCdim}(\mathscr{H}) = \infty$ and $\pi_{\mathscr{H}}(m) = 2^m$.

VC dimension and growth function

An important tool for studying VC dimension is the **growth function.** Let

$$\mathscr{H}_A = \{C \cap A \,:\, C \in \mathscr{H}\}$$

and define $\pi_{\mathscr{H}} : m \to m$ by

$$\pi_{\mathscr{H}}(m) = \max\{\|\mathscr{H}_A\| \,:\, A \subseteq X, \|A\| = m\}\,.$$

Lemma (Sauer-Shelah): For any $\mathscr{H}$, either

- $\mathrm{VCdim}(\mathscr{H}) = d$ and $\pi_{\mathscr{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} = O(m^d)$,
- or else $\mathrm{VCdim}(\mathscr{H}) = \infty$ and $\pi_{\mathscr{H}}(m) = 2^m$.

VC dimension and growth function

An important tool for studying VC dimension is the **growth function.** Let

$$\mathscr{H}_A = \{C \cap A : C \in \mathscr{H}\}$$

and define $\pi_{\mathscr{H}} : m \to m$ by

$$\pi_{\mathscr{H}}(m) = \max\{\|\mathscr{H}_A\| : A \subseteq X, \|A\| = m\}.$$

Lemma (Sauer-Shelah): For any $\mathscr{H}$, either

- VCdim($\mathscr{H}$) $= d$ and $\pi_{\mathscr{H}}(m) \leq \sum_{i=0}^{d} \left(\begin{array}{c} m \\ i \end{array} \right) = O(m^d)$,
- or else VCdim($\mathscr{H}$) $= \infty$ and $\pi_{\mathscr{H}}(m) = 2^m$.

VC dimension of a perceptron

A *perceptron* P_n with n real-valued inputs $x_1, \ldots, x_n$ gives a binary output

$$P_n(x_1, \ldots, x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta \geq 0; \\ 0, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta < 0. \end{cases}$$

As we train the parameters $b_1, \ldots, b_n, \theta$, the perceptron learns a concept $C \in \mathscr{H}_{P_n}$ bounded by a hyperplane in $\mathbb{R}^n$.

$S \subseteq \mathbb{R}^n$ is shattered by $\mathscr{H}_{P_n}$ iff every subset of S is separable by a hyperplane iff S is affine independent, so

$$\mathrm{VCdim}(\mathscr{H}_{P_n}) = n + 1.$$

VC dimension of a perceptron

A *perceptron* P_n with n real-valued inputs $x_1, \ldots, x_n$ gives a binary output

$$P_n(x_1, \ldots, x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta \geq 0; \\ 0, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta < 0. \end{cases}$$

As we train the parameters $b_1, \ldots, b_n, \theta$, the perceptron learns a concept $C \in \mathscr{H}_{P_n}$ bounded by a hyperplane in $\mathbb{R}^n$.

$S \subseteq \mathbb{R}^n$ is shattered by $\mathscr{H}_{P_n}$ iff every subset of S is separable by a hyperplane iff S is affine independent, so

$$\mathrm{VCdim}(\mathscr{H}_{P_n}) = n + 1.$$

VC dimension of a perceptron

A *perceptron* P_n with n real-valued inputs $x_1, \ldots, x_n$ gives a binary output

$$P_n(x_1, \ldots, x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta \geq 0; \\ 0, & \text{if } \sum_{i=1}^{n} b_i x_i + \theta < 0. \end{cases}$$

As we train the parameters $b_1, \ldots, b_n, \theta$, the perceptron learns a concept $C \in \mathscr{H}_{P_n}$ bounded by a hyperplane in $\mathbb{R}^n$.

$S \subseteq \mathbb{R}^n$ is shattered by $\mathscr{H}_{P_n}$ iff every subset of S is separable by a hyperplane iff S is affine independent, so

$$\mathrm{VCdim}(\mathscr{H}_{P_n}) = n + 1.$$

VC dimension and neural nets

Say N is a feed-forward neural net with n real-valued inputs, W real-valued parameters, and a binary output.

There is a corresponding concept class $\mathscr{H}_N$ of all binary concepts N can "learn" – so what is its VC-dimension?

Theorem: If the activation functions σ are step functions, then

$$\mathrm{VCdim}(\mathscr{H}_N) < 2W \log_2 \left(\frac{2W}{\log(2)} \right) = O(W \log(W)).$$

If the activation functions are sigmoid ($\sigma(z) = \frac{1}{1+e^{-z}}$), then

$$\mathrm{VCdim}(\mathscr{H}_N) = O(W^4).$$

(Karpinski and Macintyre, '95)

Corollary: neural networks can learn things.

VC dimension and neural nets

Say N is a feed-forward neural net with n real-valued inputs, W real-valued parameters, and a binary output.

There is a corresponding concept class $\mathscr{H}_N$ of all binary concepts N can "learn" – so what is its VC-dimension?

Theorem: If the activation functions σ are step functions, then

$$\mathrm{VCdim}(\mathscr{H}_N) < 2W \log_2\left(\frac{2W}{\log(2)}\right) = O(W \log(W)).$$

If the activation functions are sigmoid ($\sigma(z) = \frac{1}{1+e^{-z}}$), then

$$\mathrm{VCdim}(\mathscr{H}_N) = O(W^4).$$

(Karpinski and Macintyre, '95)

Corollary: neural networks can learn things.

VC dimension and neural nets

Say N is a feed-forward neural net with n real-valued inputs, W real-valued parameters, and a binary output.

There is a corresponding concept class $\mathscr{H}_N$ of all binary concepts N can "learn" – so what is its VC-dimension?

Theorem: If the activation functions σ are step functions, then

$$\mathrm{VCdim}(\mathscr{H}_N) < 2W\log_2\left(\frac{2W}{\log(2)}\right) = O(W\log(W)).$$

If the activation functions are sigmoid ($\sigma(z) = \frac{1}{1+e^{-z}}$), then

$$\mathrm{VCdim}(\mathscr{H}_N) = O(W^4).$$

(Karpinski and Macintyre, '95)

Corollary: neural networks can learn things.

VC dimension and neural nets

Say N is a feed-forward neural net with n real-valued inputs, W real-valued parameters, and a binary output.

There is a corresponding concept class $\mathscr{H}_N$ of all binary concepts N can "learn" – so what is its VC-dimension?

Theorem: If the activation functions σ are step functions, then

$$\mathrm{VCdim}(\mathscr{H}_N) < 2W \log_2\left(\frac{2W}{\log(2)}\right) = O(W \log(W)).$$

If the activation functions are sigmoid ($\sigma(z) = \frac{1}{1+e^{-z}}$), then

$$\mathrm{VCdim}(\mathscr{H}_N) = O(W^4).$$

(Karpinski and Macintyre, '95)

Corollary: neural networks can learn things.

VC dimension and neural nets

Say N is a feed-forward neural net with n real-valued inputs, W real-valued parameters, and a binary output.

There is a corresponding concept class $\mathscr{H}_N$ of all binary concepts N can "learn" – so what is its VC-dimension?

Theorem: If the activation functions σ are step functions, then

$$\mathrm{VCdim}(\mathscr{H}_N) < 2W \log_2 \left(\frac{2W}{\log(2)} \right) = O(W \log(W)).$$

If the activation functions are sigmoid ($\sigma(z) = \frac{1}{1+e^{-z}}$), then

$$\mathrm{VCdim}(\mathscr{H}_N) = O(W^4).$$

(Karpinski and Macintyre, '95)

Corollary: neural networks can learn things.

Sample complexity bounds for Neural Nets

Taigman et al. 2014: achieved 97.35% accuracy ($\epsilon = 0.0265$) on facial recognition task using network with $W \approx 1.2 \times 10^7$ parameters on a training set of 4×10^6 samples.

For a linear threshold network N of such a size,

$$d = \mathrm{VCdim}(\mathscr{H}_N) \leq 7.1 \times 10^9.$$

The bound by Blumer et al. guarantees 97.35% accuracy only if

$$m(\delta, \epsilon) \geq \frac{8d}{\epsilon} \log_2 \left(\frac{13}{\epsilon} \right) \approx 2 \times 10^{13}$$

samples.

Sample complexity bounds for Neural Nets

Taigman et al. 2014: achieved 97.35% accuracy ($\epsilon = 0.0265$) on facial recognition task using network with $W \approx 1.2 \times 10^7$ parameters on a training set of 4×10^6 samples.

For a linear threshold network N of such a size,

$$d = \mathrm{VCdim}(\mathscr{H}_N) \leq 7.1 \times 10^9.$$

The bound by Blumer et al. guarantees 97.35% accuracy only if

$$m(\delta, \epsilon) \geq \frac{8d}{\epsilon} \log_2 \left(\frac{13}{\epsilon} \right) \approx 2 \times 10^{13}$$

samples.

Sample complexity bounds for Neural Nets

Taigman et al. 2014: achieved 97.35% accuracy ($\epsilon = 0.0265$) on facial recognition task using network with $W \approx 1.2 \times 10^7$ parameters on a training set of 4×10^6 samples.

For a linear threshold network N of such a size,

$$d = \mathrm{VCdim}(\mathscr{H}_N) \leq 7.1 \times 10^9.$$

The bound by Blumer et al. guarantees 97.35% accuracy only if

$$m(\delta, \epsilon) \geq \frac{8d}{\epsilon} \log_2 \left(\frac{13}{\epsilon} \right) \approx 2 \times 10^{13}$$

samples.

VC dimension and logic (Laskowski's observation)

Say $\mathscr{L}$ is a first-order language, $\mathfrak{M}$ is an $\mathscr{L}$-structure, and $\varphi(x_1, \ldots, x_n; y_1, \ldots, y_m)$ is an $\mathscr{L}$-formula in first-order logic.

We may define a concept class

$$\mathscr{H}_{\varphi(\overline{x};\overline{y})} = \{\varphi(M^n; \overline{b}) : \overline{b} \in M^m\}$$

where

$$\varphi(M^n; \overline{b}) = \{\overline{a} \in M^n : \mathfrak{M} \models \varphi(\overline{a}; \overline{b})\}.$$

Then **no** $\mathscr{H}_{\varphi(\overline{x};\overline{y})}$ shatters arbitrarily large finite subsets of M^n iff **no** $\varphi(\overline{x}; \overline{y})$ has the **independence property**, or "$\mathfrak{M}$ is NIP."

VC dimension and logic (Laskowski's observation)

Say $\mathscr{L}$ is a first-order language, $\mathfrak{M}$ is an $\mathscr{L}$-structure, and $\varphi(x_1, \ldots, x_n; y_1, \ldots, y_m)$ is an $\mathscr{L}$-formula in first-order logic.

We may define a concept class

$$\mathscr{H}_{\varphi(\overline{x};\overline{y})} = \{\varphi(M^n; \overline{b}) \ : \ \overline{b} \in M^m\}$$

where

$$\varphi(M^n; \overline{b}) = \{\overline{a} \in M^n \ : \ \mathfrak{M} \models \varphi(\overline{a}; \overline{b})\}.$$

Then **no** $\mathscr{H}_{\varphi(\overline{x};\overline{y})}$ shatters arbitrarily large finite subsets of M^n iff **no** $\varphi(\overline{x}; \overline{y})$ has the **independence property**, or "$\mathfrak{M}$ is NIP."

VC dimension and logic (Laskowski's observation)

Say $\mathscr{L}$ is a first-order language, $\mathfrak{M}$ is an $\mathscr{L}$-structure, and $\varphi(x_1, \ldots, x_n; y_1, \ldots, y_m)$ is an $\mathscr{L}$-formula in first-order logic.

We may define a concept class

$$\mathscr{H}_{\varphi(\overline{x};\overline{y})} = \{\varphi(M^n; \overline{b}) \,:\, \overline{b} \in M^m\}$$

where

$$\varphi(M^n; \overline{b}) = \{\overline{a} \in M^n \,:\, \mathfrak{M} \models \varphi(\overline{a}; \overline{b})\}.$$

Then **no** $\mathscr{H}_{\varphi(\overline{x};\overline{y})}$ shatters arbitrarily large finite subsets of M^n iff **no** $\varphi(\overline{x}; \overline{y})$ has the **independence property**, or "$\mathfrak{M}$ is NIP."

Applications

Model theorists know many interesting structures $\mathfrak{M}$ with NIP, e.g.:

Theorem (Alex Wilkie): $\mathfrak{R} = (\mathbb{R}; +, \cdot, \leq, \exp)$ has NIP (the ordered field of real numbers with operation $x \mapsto e^x$ added).

Corollary 1: If $\mathscr{H} = \{C_{\overline{b}} : \overline{b} \in \mathbb{R}^m\}$ is a parametrized family of regions in $\mathbb{R}^n$ defined by a finite Boolean combination of polynomial and exponential inequalities of a fixed form, then $\text{VCdim}(\mathscr{H}) < \infty$, hence $\mathscr{H}$ is PAC learnable.

Corollary 2: If $\sigma : \mathbb{R} \to \mathbb{R}$ is defined piecewise by a finite number of applications of $+, \cdot$, division, and exponentiation, and

$$\mathscr{H} = \text{all } f : \mathbb{R}^n \to \{0,1\} \text{ computable by a NN with } W \text{ weights and activation function } \sigma$$

then $\mathscr{H}$ is PAC learnable.

Applications

Model theorists know many interesting structures $\mathfrak{M}$ with NIP, e.g.:

Theorem (Alex Wilkie): $\mathfrak{R} = (\mathbb{R}; +, \cdot, \leq, \exp)$ has NIP (the ordered field of real numbers with operation $x \mapsto e^x$ added).

Corollary 1: If $\mathscr{H} = \{C_{\overline{b}} : \overline{b} \in \mathbb{R}^m\}$ is a parametrized family of regions in $\mathbb{R}^n$ defined by a finite Boolean combination of polynomial and exponential inequalities of a fixed form, then $\mathrm{VCdim}(\mathscr{H}) < \infty$, hence $\mathscr{H}$ is PAC learnable.

Corollary 2: If $\sigma : \mathbb{R} \to \mathbb{R}$ is defined piecewise by a finite number of applications of $+, \cdot$, division, and exponentiation, and

$$\mathscr{H} = \text{all } f : \mathbb{R}^n \to \{0,1\} \text{ computable by a NN with } W \text{ weights and activation function } \sigma$$

then $\mathscr{H}$ is PAC learnable.

Applications

Model theorists know many interesting structures $\mathfrak{M}$ with NIP, e.g.:

Theorem (Alex Wilkie): $\mathfrak{R} = (\mathbb{R}; +, \cdot, \leq, \exp)$ has NIP (the ordered field of real numbers with operation $x \mapsto e^x$ added).

Corollary 1: If $\mathscr{H} = \{C_{\overline{b}} : \overline{b} \in \mathbb{R}^m\}$ is a parametrized family of regions in $\mathbb{R}^n$ defined by a finite Boolean combination of polynomial and exponential inequalities of a fixed form, then $\mathrm{VCdim}(\mathscr{H}) < \infty$, hence $\mathscr{H}$ is PAC learnable.

Corollary 2: If $\sigma : \mathbb{R} \to \mathbb{R}$ is defined piecewise by a finite number of applications of $+, \cdot,$ division, and exponentiation, and

$$\mathscr{H} = \text{all } f : \mathbb{R}^n \to \{0,1\} \text{ computable by a NN with } W \text{ weights and activation function } \sigma$$

then $\mathscr{H}$ is PAC learnable.

Applications

Model theorists know many interesting structures $\mathfrak{M}$ with NIP, e.g.:

Theorem (Alex Wilkie): $\mathfrak{R} = (\mathbb{R}; +, \cdot, \leq, \exp)$ has NIP (the ordered field of real numbers with operation $x \mapsto e^x$ added).

Corollary 1: If $\mathscr{H} = \{C_{\overline{b}} : \overline{b} \in \mathbb{R}^m\}$ is a parametrized family of regions in $\mathbb{R}^n$ defined by a finite Boolean combination of polynomial and exponential inequalities of a fixed form, then $\text{VCdim}(\mathscr{H}) < \infty$, hence $\mathscr{H}$ is PAC learnable.

Corollary 2: If $\sigma : \mathbb{R} \to \mathbb{R}$ is defined piecewise by a finite number of applications of $+, \cdot$, division, and exponentiation, and

$$\mathscr{H} = \text{all } f : \mathbb{R}^n \to \{0,1\} \text{ computable by a NN with } W \text{ weights and activation function } \sigma$$

then $\mathscr{H}$ is PAC learnable.

Sample compression

PAC learnability is also related to **compressibility** of samples.

$\mathscr{H}$ has a **compression scheme of size** $k + b$ if for any $C \in \mathscr{H}$, the labels of any C-labeled sample S can be reconstructed from a size-k subsample $S' \subseteq S$ and b bits of extra information.

Example: If $\mathscr{H}$ = all closed intervals $[a, b] \subseteq \mathbb{R}$, the labels of a sample S can be reconstructed from a size-2 subsample $S' \subseteq S$ plus two extra bits of information.

(If S contains at least two positively-labeled instance, let S' contain the least and the greatest positively-labeled instances. Use 2 extra bits of information to encode case when S contains ≤ 1 positive instance.)

Sample compression

PAC learnability is also related to **compressibility** of samples.

$\mathscr{H}$ has a **compression scheme of size** $k + b$ if for any $C \in \mathscr{H}$, the labels of any C-labeled sample S can be reconstructed from a size-k subsample $S' \subseteq S$ and b bits of extra information.

Example: If $\mathscr{H}$ = all closed intervals $[a, b] \subseteq \mathbb{R}$, the labels of a sample S can be reconstructed from a size-2 subsample $S' \subseteq S$ plus two extra bits of information.

(If S contains at least two positively-labeled instance, let S' contain the least and the greatest positively-labeled instances. Use 2 extra bits of information to encode case when S contains ≤ 1 positive instance.)

Sample compression

PAC learnability is also related to **compressibility** of samples.

$\mathscr{H}$ has a **compression scheme of size** $k + b$ if for any $C \in \mathscr{H}$, the labels of any C-labeled sample S can be reconstructed from a size-k subsample $S' \subseteq S$ and b bits of extra information.

Example: If $\mathscr{H}$ = all closed intervals $[a, b] \subseteq \mathbb{R}$, the labels of a sample S can be reconstructed from a size-2 subsample $S' \subseteq S$ plus two extra bits of information.

(If S contains at least two positively-labeled instance, let S' contain the least and the greatest positively-labeled instances. Use 2 extra bits of information to encode case when S contains ≤ 1 positive instance.)

Sample compression

PAC learnability is also related to **compressibility** of samples.

$\mathscr{H}$ has a **compression scheme of size** $k + b$ if for any $C \in \mathscr{H}$, the labels of any C-labeled sample S can be reconstructed from a size-k subsample $S' \subseteq S$ and b bits of extra information.

Example: If $\mathscr{H}$ = all closed intervals $[a, b] \subseteq \mathbb{R}$, the labels of a sample S can be reconstructed from a size-2 subsample $S' \subseteq S$ plus two extra bits of information.

(If S contains at least two positively-labeled instance, let S' contain the least and the greatest positively-labeled instances. Use 2 extra bits of information to encode case when S contains ≤ 1 positive instance.)

Warmuth's conjecture

Theorem (Littlestone and Warmuth, '86) If $\mathscr{H}$ has a compression scheme of size k, then the growth function $\pi_{\mathscr{H}}(m)$ is $O(m^k)$. In particular, $\mathscr{H}$ is PAC learnable.

Warmuth's Conjecture: Any PAC learnable class has a compression scheme of finte size.

Theorem (Moran and Yehudayoff, 2015) Warmuth's Conjecture is true; in fact,

$$\mathrm{VCdim}(\mathscr{H}) = d \Rightarrow \mathscr{H} \text{ has compression scheme of size } 2^{O(d)}.$$

Proof uses: existence of good approximation of VC classes by finite samples, Von Neumann's Minimax Theorem.

Warmuth's conjecture

Theorem (Littlestone and Warmuth, '86) If $\mathscr{H}$ has a compression scheme of size k, then the growth function $\pi_{\mathscr{H}}(m)$ is $O(m^k)$. In particular, $\mathscr{H}$ is PAC learnable.

Warmuth's Conjecture: Any PAC learnable class has a compression scheme of finte size.

Theorem (Moran and Yehudayoff, 2015) Warmuth's Conjecture is true; in fact,

$$\mathrm{VCdim}(\mathscr{H}) = d \Rightarrow \mathscr{H} \text{ has compression scheme of size } 2^{O(d)}.$$

Proof uses: existence of good approximation of VC classes by finite samples, Von Neumann's Minimax Theorem.

Warmuth's conjecture

Theorem (Littlestone and Warmuth, '86) If $\mathscr{H}$ has a compression scheme of size k, then the growth function $\pi_{\mathscr{H}}(m)$ is $O(m^k)$. In particular, $\mathscr{H}$ is PAC learnable.

Warmuth's Conjecture: Any PAC learnable class has a compression scheme of finte size.

Theorem (Moran and Yehudayoff, 2015) Warmuth's Conjecture is true; in fact,

$$\mathrm{VCdim}(\mathscr{H}) = d \Rightarrow \mathscr{H} \text{ has compression scheme of size } 2^{O(d)}.$$

Proof uses: existence of good approximation of VC classes by finite samples, Von Neumann's Minimax Theorem.

Warmuth's conjecture

Theorem (Littlestone and Warmuth, '86) If $\mathscr{H}$ has a compression scheme of size k, then the growth function $\pi_{\mathscr{H}}(m)$ is $O(m^k)$. In particular, $\mathscr{H}$ is PAC learnable.

Warmuth's Conjecture: Any PAC learnable class has a compression scheme of finte size.

Theorem (Moran and Yehudayoff, 2015) Warmuth's Conjecture is true; in fact,

$$\text{VCdim}(\mathscr{H}) = d \Rightarrow \mathscr{H} \text{ has compression scheme of size } 2^{O(d)}.$$

Proof uses: existence of good approximation of VC classes by finite samples, Von Neumann's Minimax Theorem.

Warmuth's conjecture

Theorem (Littlestone and Warmuth, '86) If $\mathscr{H}$ has a compression scheme of size k, then the growth function $\pi_{\mathscr{H}}(m)$ is $O(m^k)$. In particular, $\mathscr{H}$ is PAC learnable.

Warmuth's Conjecture: Any PAC learnable class has a compression scheme of finte size.

Theorem (Moran and Yehudayoff, 2015) Warmuth's Conjecture is true; in fact,

$$\text{VCdim}(\mathscr{H}) = d \Rightarrow \mathscr{H} \text{ has compression scheme of size } 2^{O(d)}.$$

Proof uses: existence of good approximation of VC classes by finite samples, Von Neumann's Minimax Theorem.

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Other learning models

- **Agnostic PAC learning:** Maybe the true concept C is not in $\mathscr{H}$, but our algorithm A chooses the $h \in \mathscr{H}$ which best fits a given finite sample. *This is guaranteed to converge if and only if $\mathscr{H}$ is PAC learnable.*
- **Efficient PAC learning:** Require A to have polynomial runtime in $1/\delta$ and $1/\epsilon$. This was considered by Valiant ('84) and Kearns-Vazirani ('94).
- **Online learning:** We guess how to classify given data points, and hope for a uniform finite bound on the number of mistakes.
- **Equivalence Query learning** (Angluin '86)
- *et cetera*

Online learnability

Again, there is a hypothesis class $\mathscr{H} \subseteq \mathscr{P}(X)$ (known to the learner) and we try to learn some $C \in \mathscr{H}$.

Points $x_1, x_2, x_3, \ldots \in X$ are chosen one at a time.

At stage i, teacher chooses x_i, then the learner must "guess" whether $x_i \in C$, and then learner is told whether she was correct.

The teacher may be evil and select tricky examples x_{i+1} depending on the learner's responses to $x_1, \ldots, x_i$.

$\mathscr{H}$ is **online learnable** if there is some algorithm A and some finite d such that applying A, the learner will never make more than d mistakes.

Online learnability

Again, there is a hypothesis class $\mathscr{H} \subseteq \mathscr{P}(X)$ (known to the learner) and we try to learn some $C \in \mathscr{H}$.

Points $x_1, x_2, x_3, \ldots \in X$ are chosen one at a time.

At stage i, teacher chooses x_i, then the learner must "guess" whether $x_i \in C$, and then learner is told whether she was correct.

The teacher may be evil and select tricky examples x_{i+1} depending on the learner's responses to $x_1, \ldots, x_i$.

$\mathscr{H}$ is **online learnable** if there is some algorithm A and some finite d such that applying A, the learner will never make more than d mistakes.

Online learnability

Again, there is a hypothesis class $\mathscr{H} \subseteq \mathscr{P}(X)$ (known to the learner) and we try to learn some $C \in \mathscr{H}$.

Points $x_1, x_2, x_3, \ldots \in X$ are chosen one at a time.

At stage i, teacher chooses x_i, then the learner must "guess" whether $x_i \in C$, and then learner is told whether she was correct.

The teacher may be evil and select tricky examples x_{i+1} depending on the learner's responses to $x_1, \ldots, x_i$.

$\mathscr{H}$ is **online learnable** if there is some algorithm A and some finite d such that applying A, the learner will never make more than d mistakes.

Online learnability

Again, there is a hypothesis class $\mathscr{H} \subseteq \mathscr{P}(X)$ (known to the learner) and we try to learn some $C \in \mathscr{H}$.

Points $x_1, x_2, x_3, \ldots \in X$ are chosen one at a time.

At stage i, teacher chooses x_i, then the learner must "guess" whether $x_i \in C$, and then learner is told whether she was correct.

The teacher may be evil and select tricky examples x_{i+1} depending on the learner's responses to $x_1, \ldots, x_i$.

$\mathscr{H}$ is **online learnable** if there is some algorithm A and some finite d such that applying A, the learner will never make more than d mistakes.

Online learnability

Again, there is a hypothesis class $\mathscr{H} \subseteq \mathscr{P}(X)$ (known to the learner) and we try to learn some $C \in \mathscr{H}$.

Points $x_1, x_2, x_3, \ldots \in X$ are chosen one at a time.

At stage i, teacher chooses x_i, then the learner must "guess" whether $x_i \in C$, and then learner is told whether she was correct.

The teacher may be evil and select tricky examples x_{i+1} depending on the learner's responses to $x_1, \ldots, x_i$.

$\mathscr{H}$ is **online learnable** if there is some algorithm A and some finite d such that applying A, the learner will never make more than d mistakes.

Examples of online learning

Example 1: Fix d, and let $\mathscr{H}_d \subseteq \mathscr{P}(\mathbb{R}^d)$ be all graphs of degree-d polynomials.

$\mathscr{H}_d$ is online learnable: learner should always guess that a sample point x_i is **not** on the graph, until she has found $d+1$ positive examples, after which she will know the polynomial (Lagrange's Interpolation Theorem). She will make no more than $d+1$ mistakes.

Example 2: The class $\mathscr{H}$ of all closed intervals $[a, b]$ in $\mathbb{R}$ is **not** online learnable.

Given any algorithm A, we can recursively construct points $x_1, x_2, \ldots \in \mathbb{R}$ such that A will always make the wrong guess for x_i.

Examples of online learning

Example 1: Fix d, and let $\mathscr{H}_d \subseteq \mathscr{P}(\mathbb{R}^d)$ be all graphs of degree-d polynomials.

$\mathscr{H}_d$ is online learnable: learner should always guess that a sample point x_i is **not** on the graph, until she has found $d+1$ positive examples, after which she will know the polynomial (Lagrange's Interpolation Theorem). She will make no more than $d+1$ mistakes.

Example 2: The class $\mathscr{H}$ of all closed intervals $[a, b]$ in $\mathbb{R}$ is **not** online learnable.

Given any algorithm A, we can recursively construct points $x_1, x_2, \ldots \in \mathbb{R}$ such that A will always make the wrong guess for x_i.

Examples of online learning

Example 1: Fix d, and let $\mathscr{H}_d \subseteq \mathscr{P}(\mathbb{R}^d)$ be all graphs of degree-d polynomials.

$\mathscr{H}_d$ is online learnable: learner should always guess that a sample point x_i is **not** on the graph, until she has found $d+1$ positive examples, after which she will know the polynomial (Lagrange's Interpolation Theorem). She will make no more than $d+1$ mistakes.

Example 2: The class $\mathscr{H}$ of all closed intervals $[a, b]$ in $\mathbb{R}$ is **not** online learnable.

Given any algorithm A, we can recursively construct points $x_1, x_2, \ldots \in \mathbb{R}$ such that A will always make the wrong guess for x_i.

Examples of online learning

Example 1: Fix d, and let $\mathscr{H}_d \subseteq \mathscr{P}(\mathbb{R}^d)$ be all graphs of degree-d polynomials.

$\mathscr{H}_d$ is online learnable: learner should always guess that a sample point x_i is **not** on the graph, until she has found $d+1$ positive examples, after which she will know the polynomial (Lagrange's Interpolation Theorem). She will make no more than $d+1$ mistakes.

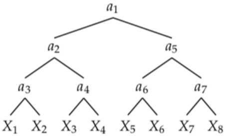
Example 2: The class $\mathscr{H}$ of all closed intervals $[a, b]$ in $\mathbb{R}$ is **not** online learnable.

Given any algorithm A, we can recursively construct points $x_1, x_2, \ldots \in \mathbb{R}$ such that A will always make the wrong guess for x_i.

Littlestone dimension

Theorem (Littlestone '88): $\mathscr{H}$ is online learnable with at most d mistakes if and only if $Ldim(\mathscr{H}) \leq d$, where $Ldim(\mathscr{H})$ is the maximum height of a binary $\mathscr{T}$ such that

- internal notes of $\mathscr{T}$ are labeled by elements of X;
- leaves of $\mathscr{T}$ are labeled by concepts in $\mathscr{H}$; and
- leaf X_i is right-below node a_j iff $a_j \in X_i$.

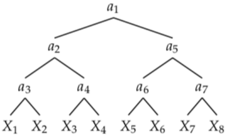


X_4 contains a_4 and a_2, but $a_1 \notin X_4$. $\mathrm{Ldim}(\{X_1, \ldots, X_8\}) = 3$.

Littlestone dimension

Theorem (Littlestone '88): $\mathscr{H}$ is online learnable with at most d mistakes if and only if $Ldim(\mathscr{H}) \leq d$, where $Ldim(\mathscr{H})$ is the maximum height of a binary $\mathscr{T}$ such that

- internal notes of $\mathscr{T}$ are labeled by elements of X;
- leaves of $\mathscr{T}$ are labeled by concepts in $\mathscr{H}$; and
- leaf X_i is right-below node a_j iff $a_j \in X_i$.



X_4 contains a_4 and a_2, but $a_1 \notin X_4$. $\text{Ldim}(\{X_1, \ldots, X_8\}) = 3$.

Online learnability and logic

Chase and Freitag (2018): Just as first-order structures in which all definable classes are PAC-learnable are NIP, structures in which all definable classes are online learnable are characterized by **stability.**

(Roughly speaking, $\mathscr{M}$ is stable if there is no infinite linear oder definable on its elements.)

We know many examples of stable infinite structures, in which all definable classes are online learnable:

- Algebraically closed fields;
- Abelian groups;
- Differentially closes fields of characteristic 0;
- (etc.)

Online learnability and logic

Chase and Freitag (2018): Just as first-order structures in which all definable classes are PAC-learnable are NIP, structures in which all definable classes are online learnable are characterized by **stability.**

(Roughly speaking, $\mathscr{M}$ is stable if there is no infinite linear oder definable on its elements.)

We know many examples of stable infinite structures, in which all definable classes are online learnable:

- Algebraically closed fields;
- Abelian groups;
- Differentially closes fields of characteristic 0;
- (etc.)

Online learnability and logic

Chase and Freitag (2018): Just as first-order structures in which all definable classes are PAC-learnable are NIP, structures in which all definable classes are online learnable are characterized by **stability.**

(Roughly speaking, $\mathscr{M}$ is stable if there is no infinite linear oder definable on its elements.)

We know many examples of stable infinite structures, in which all definable classes are online learnable:

- Algebraically closed fields;
- Abelian groups;
- Differentially closes fields of characteristic 0;
- (etc.)

Future directions

- PAC learnability must work for any sample distribution μ. But if we know something about μ, can we get better learning bounds?
- What are optimal learning bounds for neural nets?
- Calculate precise VC-dimension and growth functions in interesting examples (Abelian groups, NIP fields, ...)
- Model theory (logic) studies many notions of "tameness" for first-order theories. Do these correspond to other interesting learning models?
- Applications of compressibility of PAC classes to logic. (E.g. 2019 work by Eshel and Kaplan: proof of Warmuth's Conjecture implies "uniform definability of types over finite sets for NIP formulas.")

Future directions

- PAC learnability must work for any sample distribution μ. But if we know something about μ, can we get better learning bounds?
- What are optimal learning bounds for neural nets?
- Calculate precise VC-dimension and growth functions in interesting examples (Abelian groups, NIP fields, ...)
- Model theory (logic) studies many notions of "tameness" for first-order theories. Do these correspond to other interesting learning models?
- Applications of compressibility of PAC classes to logic. (E.g. 2019 work by Eshel and Kaplan: proof of Warmuth's Conjecture implies "uniform definability of types over finite sets for NIP formulas.")

Future directions

- PAC learnability must work for any sample distribution μ. But if we know something about μ, can we get better learning bounds?
- What are optimal learning bounds for neural nets?
- Calculate precise VC-dimension and growth functions in interesting examples (Abelian groups, NIP fields, ...)
- Model theory (logic) studies many notions of "tameness" for first-order theories. Do these correspond to other interesting learning models?
- Applications of compressibility of PAC classes to logic. (E.g. 2019 work by Eshel and Kaplan: proof of Warmuth's Conjecture implies "uniform definability of types over finite sets for NIP formulas.")

Future directions

- PAC learnability must work for any sample distribution μ. But if we know something about μ, can we get better learning bounds?
- What are optimal learning bounds for neural nets?
- Calculate precise VC-dimension and growth functions in interesting examples (Abelian groups, NIP fields, ...)
- Model theory (logic) studies many notions of "tameness" for first-order theories. Do these correspond to other interesting learning models?
- Applications of compressibility of PAC classes to logic. (E.g. 2019 work by Eshel and Kaplan: proof of Warmuth's Conjecture implies "uniform definability of types over finite sets for NIP formulas.")

Future directions

- PAC learnability must work for any sample distribution μ. But if we know something about μ, can we get better learning bounds?
- What are optimal learning bounds for neural nets?
- Calculate precise VC-dimension and growth functions in interesting examples (Abelian groups, NIP fields, ...)
- Model theory (logic) studies many notions of “tameness” for first-order theories. Do these correspond to other interesting learning models?
- Applications of compressibility of PAC classes to logic. (E.g. 2019 work by Eshel and Kaplan: proof of Warmuth’s Conjecture implies “uniform definability of types over finite sets for NIP formulas.”)

Selected bibliography

- "Learnability and the Vapnik-Chervonenkis dimension," Blumer, Ehrenfeucht, Haussler and Warmuth, *Journal of the Asssociation for Computing Machinery* **36**, 929–965 (1989)
- "Model theory and machine learning" (preprint), Hunter Chase and James Freitag, arXiv:1801.06566v1 (2018).
- "On uniform definability of types over finite sets for NIP formulas" (preprint), Eshel and Kaplan, arXiv:1904.10336 (2019).
- "Sample compression, learnability, and the Vapnik-Chervonenkis dimension," Floyd and Warmuth, *Machine Learning* **21**, 269–304 (1995).
- "Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks," Karpinski and Macintyre, *Journal of Computer and System Sciences* **54**, 169–176 (1997).
- "Sample compression schemes for VC classes," (preprint) Shay Moran and Amir Yehudayaoff, arXiv:1503.06960v2 (2015).
- *Understanding Machine Learning: From Theory to Algorithms*, Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press, 2014.