Learnability and the Vapnik-Chervonenkis Dimension

Liangjie Hong
lih307@cse.lehigh.edu

Department of Computer Science & Engineering, Lehigh University
April 10th, 2012

Overview

Main Contributions

- Explore relationships between PAC learning and VC dimension
  - First in the literature
- Bound sample size of PAC learning by VC dimension
  - Infinite case
  - Another bound
- Introduce polynomial learnability, if learning is feasible
  - When VC dimension is finite
- Introduce Occam's Razor, if learning is not feasible
  - When VC dimension is infinite
  - Prefer simpler hypothesis

Basic Setting & Notations

- A dataset: X
  1. all possible data items, usually countably infinite
  2. distributed according to P
- A *concept* class: C
  1. has finite/infinite number of concept c_i
  2. each c_i partitions the dataset into two parts: 1 and 0
  3. unknown and to be learned
- A hypothesis space: H
  1. usually in the same space of C
  2. elements in H are called *hypotheses*
  3. our approximation to C

Basic Setting & Notations (cont'd)

- A sample of size m is
  1. choose m data items from X
  2. choose c from C:
     Example: $\text{sam}_c(\bar{x}) = (< x_1, I_c(x_1) >, \cdots, < x_m, I_c(x_m) >)$
- Sample space of C: S_C
- Learning algorithms: $A_{C,H}$
  1. all functions $A : S_C \rightarrow H$
  2. a particular algorithm A generates a $h \in H$
     Example: $(< x_1, I_h(x_1) >, \cdots, < x_m, I_h(x_m) >)$
  3. the *error* of A is defined as:
     $error_P(h) = P_{x_i \in P}[I_h(x_i) \neq I_c(x_i)]$
  4. *consistency*

Notes

- This setting is very different from classical settings.
  - No notion of training and testing at all.
- We care about the concept class C.
  - A family of problems not a single problem.
- Only about classification problems.
  - How about regression, density estimation?

Table of Contents

PAC Learnability and VC Dimension

- PAC learnability (Review)
- Vapnik-Chervonenkis dimension
- Bounding sample size in PAC learning with VC dimension

PAC Learnability

The motivation of PAC learnability:

- We want A is as accurate as possible:
  - $error_P(h)$ is small $\rightarrow$ $error_P(h) \leq \varepsilon$
- We can make this accuracy confidently:
  - $P(error_P(h) \leq \varepsilon)$ is large $\rightarrow$ $P(error_P(h) \leq \varepsilon) \geq 1 - \delta$
- We even want that A works for any P!

PAC Learnability

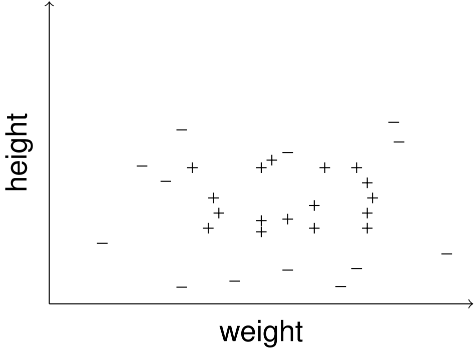
Definition:
Let $A \in A_{C,H}$ be a learning function for C (with respect to P) with sample size $m(\varepsilon, \delta)$. If A satisfies the condition that given any $\varepsilon, \delta \in [0,1]$, $P(error_P(h) > \varepsilon) \leq \delta$ for all $c \in C$, we say that C is *uniformly learnable by H under the distribution P*.

PAC Learnability

Definition:
Let $A \in A_{C,H}$ be a learning function for C (with respect to P) with sample size $m(\varepsilon, \delta)$. If A satisfies the condition that given any $\varepsilon, \delta \in [0, 1]$, $P(error_P(h) > \varepsilon) \leq \delta$ for all $c \in C$, we say that C is *uniformly learnable by H under the distribution P*.

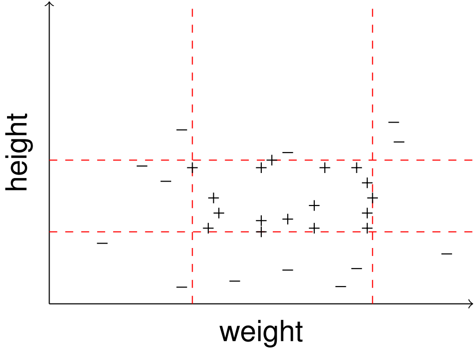
- Sample size $m(\varepsilon, \delta)$ is an integer-valued function of ε and δ.
- A is a learning function only when A is a learning function for C with all P!
- The smallest $m(\varepsilon, \delta)$ is called the *sample complexity* of A.

PAC Learnability: Example

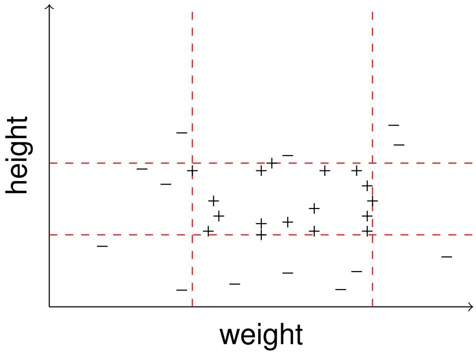


A target concept c is a rectangle.

PAC Learnability: Example



PAC Learnability: Example



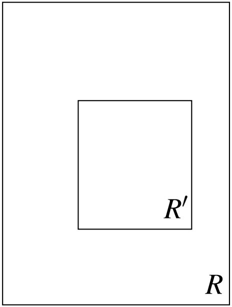
Call the learning function defined by this algorithm A.

PAC Learnability: Example

What is the sample complexity of algorithm A?

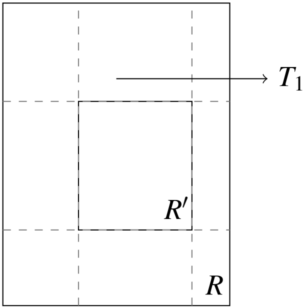
- Denote the target region as R
- Denote the learned region as R'
- Define *weight* $w(E)$ of a region E as: $w(E) = \int_{x \in E} P(x)\,dx$
- Define $error(R')$ as:
  $error(R') = w(R - R')$
- Goal:
  We want to bound $error(R') \leq \varepsilon$ with probability at least $1 - \delta$ after seeing m examples.

PAC Learnability: Example



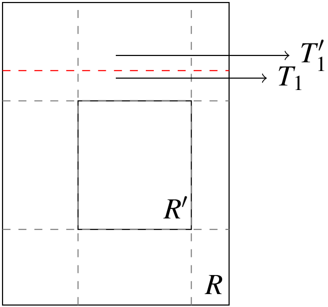
Total error is: ε.

PAC Learnability: Example



- Each strip should have error at most $\frac{\epsilon}{4}$
- Estimate $P(w(T_1) > \frac{\epsilon}{4})$

PAC Learnability: Example



- Let $w(T_1') = \frac{\varepsilon}{4}$.
- No points in T_1' appear in the sample. (why?)
- The probability of a point falls outside $T_1' = 1 - \frac{\varepsilon}{4}$.

PAC Learnability: Example

- The whole sample is outside of T_1': $[1-\frac{\varepsilon}{4}]^m$
- In other words, $P(w(T_1) > \frac{\varepsilon}{4})$ is at most $[1-\frac{\varepsilon}{4}]^m$
- Same analysis applies to four similar strips.
- By using union bound $P(A \cup B) \leq P(A) + P(B)$:
- $P(error(R') \geq \varepsilon) \leq 4[1-\frac{\varepsilon}{4}]^m \leq \delta$
- By some algebraic transformations, we can conclude:

$$m \geq \frac{4}{\varepsilon} \log \frac{4}{\delta}$$

PAC Learnability: Example

- This applies to any P.
- The sample size m is bounded.
- The growth of m is linear in $\frac{1}{\epsilon}$ and linear in $\log \frac{1}{\delta}$.

PAC Learnability

In general, for any finite concept class $|C| < \infty$, it is learnable and the learning algorithms simply need to generate consistent hypotheses with:

$$m \geq \frac{1}{\varepsilon} \log \frac{|C|}{\delta}$$

Question

How about infinite cardinality of C?

VC Dimension

Definition:
Given a nonempty concept class C and a set of points $S \in X$, $\Pi_C(S)$ denotes the **set** of all subsets of S that can be obtained by intersecting S with a concept in C:

$$\Pi_C(S) = \{(I_c(x_1), \cdots, I_c(x_m) : c \in C, x_i \in S\}$$

or we can have $\Pi_C(S) = \{S \cap c : c \in C\}$. Thus, $\Pi_C(S)$ contains positive examples of S by all possible c.

VC Dimension

Definition:
If $|\Pi_C(S)| = 2^m$, then S is considered **shattered** by C.

In other words, S is shattered by C if C realizes all possible dichotomies of S.

Shattering: Example 1

Consider as an example a finite concept class $C = \{c_1, \cdots, c_4\}$ applied to three instance vectors with the results:

	x_1	x_2	x_3
c_1	1	1	1
c_2	0	1	1
c_3	1	0	0
c_4	0	0	0

Shattering: Example 1

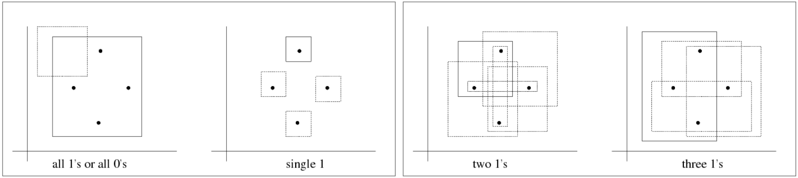
Consider as an example a finite concept class $C = \{c_1, \cdots, c_4\}$ applied to three instance vectors with the results:

	x_1	x_2	x_3
c_1	1	1	1
c_2	0	1	1
c_3	1	0	0
c_4	0	0	0

Then,

- $\Pi_C(\{x_1\})$ (shattered)
- $\Pi_C(\{x_1, x_3\})$ (shattered)
- $\Pi_C(\{x_2, x_3\})$ (not shattered)

Shattering: Example 2



Shattering with rectangles

VC Dimension

Definition:
The VC dimension of C, denoted as $VCDim(C)$, is the cardinality d of the largest set S shattered by C. If arbitrary large finite sets are shattered, then $VCDim(C) = \infty$.

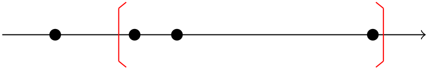
VC Dimension

Notes:

- $VCDim(C)$ is a property for the concept class C
- $VCDim(C)$ of a finite concept class $|C| < \infty$ is bounded as $\log |C|$, because $|C| \geq 2^d$

Example 1: Intervals of the real line

Let X be the real line and let C be the set of **all** intervals on X. What is $VCDim(C)$?

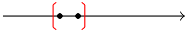
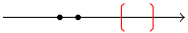
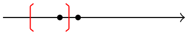
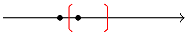


Example 1: Intervals of the real line

Let us firstly try $d = 2$.

Example 1: Intervals of the real line

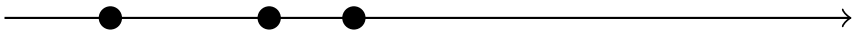
Let us firstly try $d = 2$.

Interval Placement	Labels
	1 1
	0 0
	1 0
	0 1

How about $d = 3$?

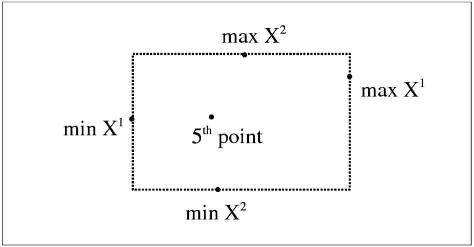
Example 1: Intervals of the real line

For $d = 3$, we cannot generate the label $\{1\ 0\ 1\}$!

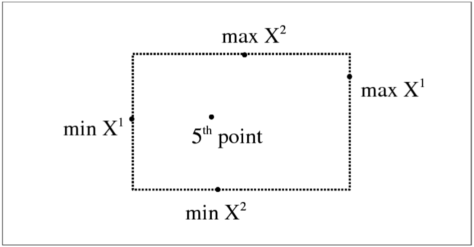


Therefore, $VCDim(C) = 2$.

Example 2: Axes-aligned rectangles in the plane

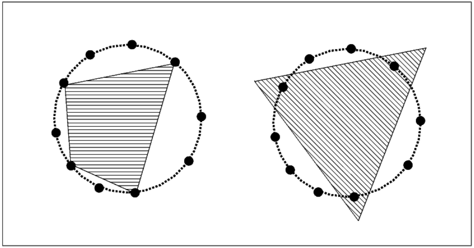


Example 2: Axes-aligned rectangles in the plane



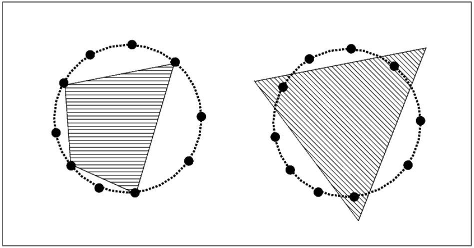
The $VCDim(C) = 4$.

Example 3: Convex polygons



Convex polygons

Example 3: Convex polygons



Convex polygons

The VC dimension is infinite.

More Conclusions about VC Dimension

- Separating hyperplanes in R^n: $n+1$.
- Union of a finite number of intervals on the line: ∞.

Bound Sample size with VC dimension

Theorem:

Let C be a nontrivial, well-behaved concept class.

1. C is uniformly learnable if and only if the VC dimension of C is finite.
2. If the VC dimension of C is d, where $d < \infty$, then:
   2.1 for $0 < \varepsilon < 1$ and sample size at least
   $$\max\left(\frac{4}{\varepsilon}\log\frac{2}{\delta}, \frac{8d}{\varepsilon}\log\frac{13}{\varepsilon}\right)$$
   any consistent function $A : S_C \rightarrow C$ is a learning function for C and
   2.2 for $0 < \varepsilon < \frac{1}{2}$ and sample size less than
   $$\max\left(\frac{1-\varepsilon}{\varepsilon}\log\frac{1}{\delta}, d(1-2(\varepsilon(1-\delta)+\delta))\right)$$
   no function $A : S_C \rightarrow H$, for any hypothesis space H, is a learning function for C.

Liangjie Hong lih307@cse.lehigh.edu

Department of Computer Science & Engineering, Lehigh University

Bound Sample size with VC dimension

Notes:

- The first part demonstrates an easier way to prove C uniformly learnable if one can show C has a finite VC dimension.
- The second part is to link sample size m with error ε, confident δ **and** VC dimension.
- Both statements do not require C finite but require $VCDim(C)$ finite!

Bound Sample size with VC dimension

Comparing bounds:

- Previous bound: $O\left(\frac{1}{\varepsilon}\left(\log\frac{1}{\delta}+\log|C|\right)\right)$
- Current bound: $O\left(\frac{1}{\varepsilon}\left(\log\frac{1}{\delta}+VCDim(C)\log\frac{1}{\varepsilon}\right)\right)$

Bound Sample size with VC dimension

Proof Sketch:

- Part 1 is automatically true if Part 2 is true.
- Part 2 is proven by:
  - Construct a special P, C and X.
  - Cannot find any A to satisfy PAC learnable conditions.

Table of Contents

Once we have sample size m, error ε and confident level δ and model complexity $VCDim(C)$, what is missing?

Once we have sample size m, error ε and confident level δ and model complexity $VCDim(C)$, what is missing?

- Computational feasibility
  - Polynomial time bound
- Control complexity of learned model
  - Occam's Razor

Polynomial Learnable

Main ideas:

- Try to define C_n is properly polynomial learnable where n is dimensionality of X.
- Depend on VC dimension of C_n grows only polynomially in n.
- This only happens when C_n has a finite VC dimension.

Polynomial Learnable

Main ideas:

- Try to define C_n is properly polynomial learnable where n is dimensionality of X.
- Depend on VC dimension of C_n grows only polynomially in n.
- This only happens when C_n has a finite VC dimension.

Redefine PAC learnable by incorporating polynomial time complexity constraint and VC dimension.

What if VC dimension is infinite?

Occam's Razor

- Define **size** be a function from C into $\mathbf{Z}^+$.
- Polynomial learnable of C is redefined by adding an additional bound $\mathbf{size}(c)$ for all $c \in C$.
- May not find the *simplest* hypotheses, but *simpler* one.

That's it!

Thank you.