PAC Learning

a discussion on the original paper by Valiant

Adrian Florea

13$^{\text{th}}$ Meetup of Papers We Love (Bucharest Chapter),
27 January 2017

Induction in an urn I

The urn

Consider an urn containing a very large number (millions) of marbles, possibly of different types. You are allowed to draw 100 marbles and asked what kinds of marbles the urn contains.

- *no assumptions* - impossible task!
- *assumption: all the marbles are of different types* - impossible task!
- *assumption: all the marbles are identical* - a single draw gives complete knowledge about all the marbles. :-)
- *assumption: 50% of the marbles are of one type* - the probability to miss that type is $(1/2)^{100} = 7.89 * 10^{-31}$

Induction in an urn II

- *assumption: there are at most 5 different marble types* - if any of the 5 types occurs with frequency $> 5\%$, the probability to miss that type is $< (1-0.05)^{100} < 0.6\%$ so the probability to miss any one of these frequent ones is $< 5 * 0.6\% = 3\%$. There can be at most 4 types with frequency $< 5\%$ so the rare types are $< 20\%$.

Remark

Even if the distribution of marble types is unknown, we can predict with 97% confidence that after 100 picks (small sample) you will have seen representatives of $\geqslant 80\%$ urn content

We needed only two assumptions:

- The Invariance Assumption: the urn do not change.
- The Learnable Regularity Assumption: there are a fixed number of marble types represented in the urn.

Definition

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

Definition

A *concept* c over X is a subset $c \subseteq X$ of the instance space.

A subset $c \subseteq X$ can be represented as $c \in 2^X$, with c as the inverse image of 1, i.e. $c\colon X \to \{0, 1\}, c(x) = 1$ if x is a *positive example* of c and $c(x) = 0$ if x is a *negative example* of c.

Definition

A *concept class* C over X is a set of concepts over X, typically with an associated representation.

Definition

A *target concept* may be any concept $c^* \in C$.

Definition

An *assignment* is a function that maps a truth value to all of its variables.

Definition

A *satisfying assignment* is when, after applying the assignment, the underlying formula simplifies to *true*.

For the sake of simplicity we can consider concepts c over $\{0,1\}^n$ whose positive examples are the satisfying assignments of Boolean formulae f_c over $\{0,1\}^n$. We can then define a concept class C by considering only f_c fulfilling certain syntactic constraints (its representation).

Definition

A *learning protocol* specifies the manner in which information is obtained from the outside.

Valiant considered two routines as part of a learning protocol:

- *EXAMPLES* routine: has no input, it returns as output a positive example x ($c^*(x) = 1$) based on a fixed and perhaps unknown probabilistic distribution determined arbitrarily by nature;
- *ORACLE*() routine: on x as input it returns 1 if $c^*(x) = 1$, or 0 if $c^*(x) = 0$.

In a real system the ORACLE() may be a human expert, a data set of past observations, etc.

Definition

A *learning algorithm* (called also *learner*) tries to infer an unknown concept (called *hypothesis*), chosen from a known concept class.

What it means for a learner to be successful?

- e.g. the learner must output a hypothesis *identical* to the target concept, or
- e.g. the hypotheses *agrees* with the target concept *most of the time*.

The learner can call the *EXAMPLES* and the *ORACLE*() routines.
The learner calls the *ORACLE*() routine over the instances received in a distribution D from the external information supply.

Definition

A set of random variables is *independent and identically distributed (i.i.d.)* if each random variable has the same *identical* probability distribution as the others and all are mutually *independent*.

The instances the learner receives from D, are independently and identically distributed (i.i.d.).

Remark

The assumption of a *fixed* distribution helps us to hope that what the learner learned from the training data will carry over to new, unseen yet, test data.

Definition

A *learning machine* consists of a learning protocol together with a learning algorithm.

After observing the sequence S of i.i.d. *training examples* of the target concept c^*, the learner L outputs the hypothesis h (its estimate of c^*) from the set H of possible hypotheses:

$$D \xrightarrow{(x_1,\ldots,x_m)} ORACLE() \xrightarrow{S\equiv((x_1,c^*(x_1)),\ldots,(x_m,c^*(x_m)))} L \xrightarrow{h\in H} H$$

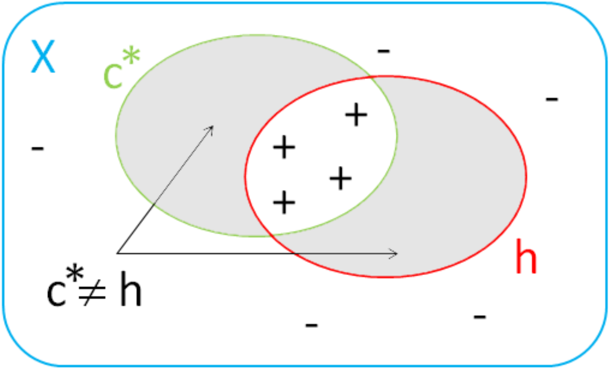
The success of L is determined by the performance of h over new i.i.d. instances drawn from X according to D.

Definition

The *true error* of h with respect to c^* and D is the probability that h will misclassify an instance drawn randomly according to D:

$$error_D(h) \equiv \Pr_{x\in D}[c^*(x) \neq h(x)]$$

- The true error is defined over D and not over S because it's about the error of using the learned hypothesis h on subsequent instances drawn from D.
- The true error depends strongly on D.



The true error cannot be observed by L. The learner can only observe the performance of h over S.

Definition

The *training error* is the fraction of training examples misclassified by h: $error_S(h) \equiv \Pr_{x \in S}[c^*(x) \neq h(x)] = \frac{1}{m}\sum_{i=1}^{m} I[c^*(x_i) \neq h(x_i)]$

As the true error depends on D, the training error depends on S.
How many training examples a learner needs to learn to output a hypothesis h?
If $error_D(h) = 0$, the learner needs $|S| = |X|$ training examples - that means *no learning*!

Remark

We can only require that the learner probably learn a hypothesis that is approximately correct!

Definition

A concept class C is *PAC-learnable* by L using H if:

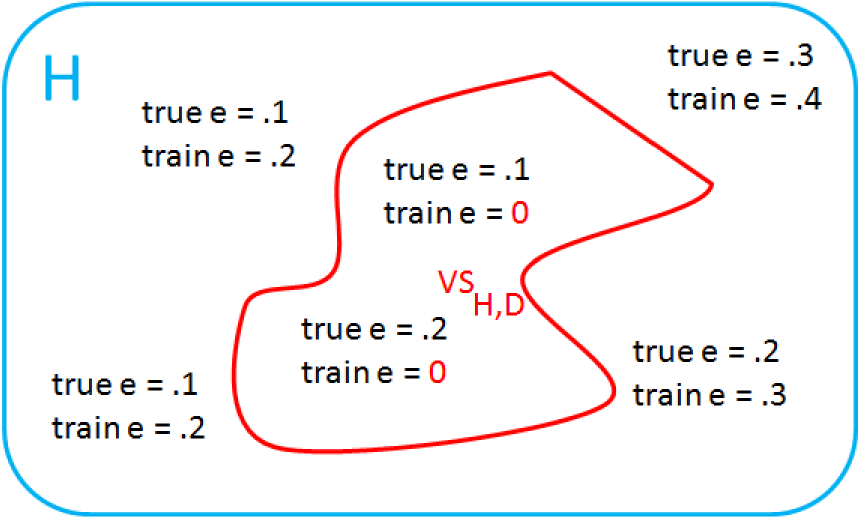
- for all $c^* \in C$
- for any D over X
- $0 < \varepsilon < \frac{1}{2}$ arbitrarily small
- $0 < \delta < \frac{1}{2}$ arbitrarily small

the learner L will, with the probability of at least $(1 - \delta)$, output a hypothesis $h \in H$ such that $error_D(h) \leq \varepsilon$, in a polynomial time in $\frac{1}{\varepsilon}$, $\frac{1}{\delta}$, $size(x \in X)$, $size(c)$.

Implicit *assumption*: $\forall c^* \in C, \exists h \in H$ s.t. $error_D(h)$ arbitrarily small

Definition

$VS_{H,D} = \{h \in H | \forall x \in D, c^*(x) = h(x)\}$ is called a *version space*


H
true e = .1
train e = .2
true e = .3
train e = .4
true e = .1
train e = 0
$VS_{H,D}$
true e = .2
train e = 0
true e = .1
train e = .2
true e = .2
train e = .3

Definition

A version space $VS_{H,D}$ is called ε*-exhausted* with respect to c^* and D, if: $\forall h \in VS_{H,D}, error_D(h) < \varepsilon$

Definition

A *consistent hypothesis* is a concept that perfectly fit the training examples.

The Theorem of ε-exhausting the version space (Haussler, 1988)

If the hypothesis space H is finite, and D is a sequence of m i.i.d. drawn examples of the target concept c^*, then for any $0 \leq \varepsilon \leq 1$, the probability that $VS_{H,D}$ is not ε-exhausted with respect to c^* is at least $|H|e^{-\varepsilon m}$

Proof: Let $h_1, h_2, ..., h_k$ be all hypotheses in H with $error_D(h_i) \geq \varepsilon, i = \overline{1, k}$. The probability that any single hypothesis h_i with $error_D(h_i) \geq \varepsilon$ is consistent with a randomly drawn example is at most $(1 - \varepsilon)$, so the probability for h_i to be consistent with all m i.i.d. examples is $(1-\varepsilon)^m$. We fail to ε-exhaust the version space iff there is such a hypothesis consistent with all m i.i.d. examples. Since $P(A \cup B) \leq P(A) + P(B)$, we have that the probability that all m examples are consistent with any of the k hypotheses is at most $k(1-\varepsilon)^m$. But $k \leq |H|$ and $1 - x \leq e^{-x}$ for $0 \leq x \leq 1$, so the probability is at most $|H|e^{-\varepsilon m}$

Corollary

$m \geq \frac{1}{\varepsilon}(ln|H| + ln(\frac{1}{\delta}))$

The number of i.i.d. examples needed to ε-exhaust a version space is logarithmic in the size of the underlying hypothesis space, independently of the target concept or the distribution over the instance space.

- Let's consider the concept class C of target concepts described by *conjunctions of Boolean literals* (Boolean variables or their negation). Is C PAC-learnable?

  If we have an algorithm that uses a polynomial time per training example, the answer is *yes* if we can show that any consistent learner requires a polynomial number of training examples.

  We have $|H| = 3^n$ because there are 3 values for a Boolean literal: the variable, its negation, and the situation when it's missing in the concept formula. So:

  $m \geq \frac{1}{\varepsilon}(n \cdot ln3 + ln(\frac{1}{\delta}))$

  Example: A consistent learner trying to learn with errors less than 0.1 with a probability of 95% a target concept described by a conjunction of up to 10 Boolean literals, requires:

  $\frac{1}{0.1}(10 \cdot ln3 + ln(\frac{1}{0.05})) = 139.8 \approx 140$

  training samples.

- Let's consider now the concept class C of all learnable concepts over X, where X is defined by n Boolean features. We have:
  $|C| = 2^{|X|}$
  $|X| = 2^n$ so $|C| = 2^{2^n}$
  To learn such a concept class, the learner must use the hypothesis space $H = C$:
  $m \geq \frac{1}{\varepsilon}(2^n \cdot ln(2) + ln(\frac{1}{\delta}))$
  exponential in n.

Bibliography I

M.J. Kearns
The Computational Complexity of Machine Learning
MIT Press, 1990

M.J. Kearns, U.V. Vazirani
An Introduction to Computational Learning Theory
MIT Press, 1994

T.M. Mitchell
Machine Learning
McGraw-Hill, 1997

B.K. Natarajan
Machine Learning. A Theoretical Approach
Morgan Kaufmann Publishers, 1991

Bibliography II

L.G. Valiant
Probably Approximately Correct. Nature's Algorithms for Learning and Prospering in a Complex World
Basic Books, 2013

J. Amsterdam
Some Philosophical Problems with Formal Learning Theory
AAAI-88 Proceedings, 580-584, 1988

D. Angluin
Learning From Noisy Examples
Machine Learning, 2:343370, 1988

C. Le Goues
A Theory of the Learnable (L.G. Valiant)
Theory Lunch Presentation, 20 May 2010

Bibliography III

D. Haussler
Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework
Artificial Intelligence, 36(2):177-221,1988

L.G. Valiant
A Theory of the Learnable
Comm. ACM, 27(11):1134–1142, 1984

L.G. Valiant
Deductive Learning
Phil. Trans. R. Soc. Lond, A 312: 441-446, 1984