Machine Learning: Foundations **Spring Semester, 2002**

Lecture 7: May 9,2002

Lecturer: Yishay Mansour Scribe: Elhanan Borenstein, Orit Kliper & Ofer Pasternak [1]

notes!

7.1 Motivation

We wish to consider the following question: How many random examples does a learning algorithm need to draw, before it has sufficient information to learn an unknown target concept chosen from the concept class C?

In lecture 3, we presented the PAC learning model and proved some lower bounds on the number of examples required for PAC learning in several hypothesis classes (of various sizes $\ln|C|$). In the previous lecture, we introduced the definition of the VC-Dimension and explored a number of examples.

In this lecture we will discuss the connection between the VC-Dimension and learning, and show how the VC-Dimension of a hypothesis class C can be used to derive the lower and upper bounds on the number of examples required. The concept of the VC-Dimension will also provide us a substitute to $\ln|C|$ as a parameter to determine the sample size, for infinite concept classes.

7.2 The PAC Model - Review

In the PAC Model we assume there exists a distribution D on the examples that the learner receives; i.e. when choosing an instance from the sample it is drawn according to D. We assume that distribution has the following characteristics:

1. Fixed throughout the learning process.
2. Unknown to the learner.
3. The instances are chosen independently.

The target concept is specified as a computable function $c(x)$, thus our instances are of the form $<x, c(x)>$. Our goal is to find a function h from H which is a good approximation of

[1] Based on scribe written by Vladimir Goldner, Yuval E. Sapir & Assaf K. Paznerin

c with respect to D, in the following sense:
Let

$$error(h, c) = \ Prob_D[c(x) \neq h(x)].$$

We would like our algorithm, with a high probability $(1 - \delta)$, to find a hypothesis h, such that this *error* is smaller than a certain threshold ϵ. That is, we require that the following hold:

$$Prob[error(h) < \epsilon] \geq 1 - \delta.$$

ϵ (the error threshold) and δ (a measure of our confidence in the outcome of the learning process), are given as parameters to the algorithm. We also assume that the hypothesis space H, includes such an hypothesis which is a good approximation of the function $c(x)$.

7.3 The VC-Dimension - Review

7.3.1 Definitions

We start with few definitions. Assume C is a concept class defined over instance space X. We can associate a concept c over X with a set (all the examples in X on which c returns a positive classification).

Let $S \subseteq X$ be a subset of X. We can define the projection of C over the subset S as follows:

Definition For each concept class C over X and for any subset $S \subseteq X$:

$$\Pi_C(S) = \{c \cap S | c \in C\}$$

Equivalently, if $S = \{x_1, \ldots, x_m\}$ then we can think of $\Pi_C(S)$ as the set of vectors and $\Pi_C(S)$ is defined by:

$$\Pi_C(S) = \{< c(x_1), \ldots, c(x_m) > | c \in C\}$$

In effect we are reducing the concept class C into the concept class $C|S$, where $S = \{x_1, ..., x_m\}$.

Clearly, The concept class $C|S$ is finite with at most 2^m different concepts (as there are at most 2^m different vectors of size m) , thus:

$$|\Pi_C(S)| \leq 2^m$$

Definition A concept class C *shatters* S if $|\Pi_C(S)| = 2^m$, or in other words a concept class shatters a set of inputs if every possible function on the input appears in $\Pi_C(S)$.

Now we are ready to define the notion of VC dimension.

Definition *VC-dim(C) (Vapnik-Chervonenkis dimension)* of C is the maximum size of a set S that is shattered by C:

$$VCdim(C) = max\{d : \exists S : |S| = d \ \ and \ \ |\Pi_C(S)| = 2^d\}.$$

If C shatters set of arbitrary large size sets (i.e such a maximum as above does not exist) we define $VCdim(C)$ to be infinity.

¿From the bound on VC-dim above ($|\Pi_C(S)| \leq 2^m$), we can also derive that for a finite class:

$$VC - dim(C) \leq \log |C|$$

7.4 Lower Bounds

7.4.1 Static Learning Model

Definition The *static learning algorithm* has the following characteristics:

- It asks for a sample of size $m(\epsilon, \delta)$
- It chooses it's hypothesis (based on the sample it got).

The main limitation is that the algorithm cannot update the number of necessary examples after it receives the sample in step 1.

In the following theorem for the static learning model, we show that if $VCdim = \infty$, then no static algorithm exists. Namely, there is no algorithm that can use a sample size which depends *only* on ϵ and δ for classes of infinite VC-dimension.

Theorem 7.1 *If a concept class C has $VCdim(C) = \infty$ then C is not learnable by any static learning algorithm.*

Proof: The proof is by contradiction. We will set ϵ and δ as follows: $\epsilon = \frac{1}{10}, \delta = \frac{1}{10}$. Let $m = m(\frac{1}{10}, \frac{1}{10})$ and let A be a static algorithm that learns C using m examples. Based on our assumption that C has infinite dimension ($VCdim(C) = \infty$), there exist $2m$ points, $z_1, ...z_{2m}$, which C shatters. Let $T = \{z_1, ..., z_{2m}\}$.

Since a PAC learning algorithm should hold for any distribution, we can define the distribution D as follows:

$$D(x) = \begin{cases} \frac{1}{2m} & x \in T \\ 0 & \text{otherwise} \end{cases}$$

Thus D is a uniform distribution over T (and 0 elsewhere).
Since T is shattered by C, then any concept is possible. Thus, we shall choose a concept c_t as follows:

$$c_t(x) = \begin{cases} \text{1 or 0 with probability } \frac{1}{2} & x \in T \\ \text{never mind} & \text{otherwise} \end{cases}$$

There exists a $c_t \in C$ which fits the random choice, because C shatters T.
Let the learning algorithm A samples m points: $B = \{x_1, ..., x_m\}$ (drawn according to D), clearly $|B| \leq m$.

Let h be the hypothesis which the algorithm A outputs. Since we have no data on the classification of the points not sampled by the algorithm, we will choose it's value to be 0 or 1, randomly. Thus, for every point, $z_i \notin B$, the probability that h makes an error is $\frac{1}{2}$, i.e.

$$Pr[c_t(z_i) \neq h(z_i)] = \frac{1}{2}$$

Computing the expected error for h gives,

$$E[error(h)] \geq m \cdot \frac{1}{2} \cdot \frac{1}{2m} = \frac{1}{4}$$

- m - at least m points were not seen (the number of points not in B)
- $\frac{1}{2}$ - Probability of error.
- $\frac{1}{2m}$ - Probability of z_i.

We still must show that with a probability $> \frac{1}{10}$ (δ) we have an hypothesis, whose error is $> \frac{1}{10}$ (ϵ). Let,

$$1 - \delta = \alpha = Prob[error(h) \leq \frac{1}{10}]$$

We can bound α as follows,

$$\frac{1}{4} \leq E[error(h)] \leq \alpha \cdot \frac{1}{10} + (1 - \alpha) \cdot 1 = 1 - \frac{9}{10} \cdot \alpha$$

Hence,

$$\alpha \leq \frac{5}{6}$$

Thus we have reached a contradiction to our assumption that $\alpha \geq \frac{9}{10}$.
Therefore, we had showed, that if $VCdim(C) = \infty$, there is no integer m such that it is sufficient to sample m examples and guarantee PAC learning with $\epsilon = \frac{1}{10}$ and $\delta = \frac{1}{10}$. □

7.4.2 Lower Bound - Feasible

We will now show, in a similar manner, that a lower bound can also be found for a finite VC-dimension.

Theorem 7.2 *If C is a class for which $VCdim(C) = d + 1$, then*

$$m(\epsilon, \delta) = \Omega(\frac{d}{\epsilon})$$

(this bound makes sense intuitively: the sample size should increase when d increases or when ϵ decreases)

Proof: Let $T = \{z_0, z_1, ..., z_d\}$, such that C shatters T. We know that such a group exists, because $VCdim(C) = d + 1$.
In this case, we will not use a uniform distribution as we did in the previous proof, but instead, we shall define D as follows:

We will set the probability of the point z_0 to $1 - 8\epsilon$, and define the probability for each z_i $(i > 0)$ to be $\frac{8\epsilon}{d}$. Formally:

$$D = \begin{cases} 1 - 8\epsilon & x = z_0 \\ \frac{8\epsilon}{d-1} & x = z_i \;\;,\;\; i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Again, we will choose c_t randomly as follows:

$$c_t(x) = \begin{cases} 1 & x = z_0 \\ 1 \text{ or } 0 \text{ with probability } \frac{1}{2} & x = z_i \;\;,\;\; i > 0 \\ \text{never mind} & \text{otherwise} \end{cases}$$

As before, we will ask, how many samples are necessary, under this distribution, in order to receive $\frac{d}{2}$ of the z's. Let's assume that the learner, "knows" only $\frac{d}{2}$ different points of z_i, $i > 0$, then:

$$E[error(h)] \geq \frac{d}{2} \cdot \frac{1}{2} \cdot \frac{8\epsilon}{d} = 2\epsilon$$

Let,

$$1 - \delta = \alpha = Prob[error(h) \leq \epsilon]$$

Then:

$$2\epsilon \leq E[error(h)] \leq \alpha\epsilon + (1 - \alpha) \cdot 1$$

And we get:

$$\alpha \leq \frac{1 - 2\epsilon}{1 - \epsilon} = 1 - \frac{\epsilon}{1 - \epsilon} \quad .$$

So if we choose $\delta = 1 - \alpha < \frac{\epsilon}{1-\epsilon}$, we will fail.

□

The question is, how many examples can we draw, and still, with a high probability, sample less than $\frac{d}{2}$ points. If we take m examples, the expectation of the number of points from $\{z_1, ..., z_n\}$ is $m \cdot \frac{8\epsilon}{d} \cdot d = 8m\epsilon$. In order to get less than $\frac{d}{2}$ points we will require $\frac{d}{2} \leq 8m\epsilon$. We get:

$$m(\epsilon, \delta) = \frac{d}{16\epsilon}$$

or

$$m(\epsilon, \delta) = \Omega(\frac{d}{\epsilon})$$

This is a lower bound on m.

7.4.3 Lower Bound - Non Feasible

We will now consider the case where the hypothesis class is non-feasible. That is, it is not guaranteed that it includes an hypothesis which is a good approximation of the target function.
We will focus on a simple case where H includes only two hypotheses.

Theorem 7.3 *If H is a hypothesis class that include only two hypotheses (and is not obliged to include the target concept c_t), then*

$$m(\epsilon, \delta) = \Omega(\frac{\log \frac{1}{\delta}}{\epsilon^2})$$

Proof: Let $H = \{h_0, h_1\}$ where $h_b(x) = b$ (that is, h_0 is a constant functions that always returns 0 and h_1 is a constant functions that always returns 1). The algorithm will simply return one of these hypotheses according to the majority on the examples.

Let D_0 and D_1 be two distributions of the form:

$$D_0 = \left\{ \begin{array}{ll} \frac{1}{2} - \gamma & <x, 1> \\ \frac{1}{2} + \gamma & <x, 0> \end{array} \right. \quad D_1 = \left\{ \begin{array}{ll} \frac{1}{2} + \gamma & <x, 1> \\ \frac{1}{2} - \gamma & <x, 0> \end{array} \right.$$

Basically, we want our learning algorithm to return h_0 if we are using the distribution D_0 and h_1 if we are using the distribution D_0. Clearly, since the two distributions are very close, if we will not have enough examples, we will not be able to determine whether the sample came form D_0 or D_1.
According to Chernoff:

$$e^{\gamma^2 m} = \delta$$

Hence:

$$m = \frac{1}{\gamma^2} \log(\frac{1}{\delta})$$

(The full details of this proof are not enclosed) □

7.5 Upper Bound

We will now turn to the important application of the VC-dimension - deriving the upper bound on the sample size. Obviously, proving an upper bound, also indicates that a learning algorithm using that bound is valid and possible.

7.5.1 ϵ-net notion

Definition Given a target concept c, we shall define the ϵ-bad concepts as the group of all the concepts that have an error larger than ϵ. Formally:

$$B_\epsilon(c) = \{h | error(h, c) > \epsilon\}$$

We show that if we perform a large enough sampling, then none of these functions will be consistent with the sample.

Definition A set of points, S, is an ϵ-net for c with respect to a distribution D, if for each concept $h \in B_\epsilon(c)$ there exists a point $x \in S$ such that $h(x) \neq c(x)$.

The important property of ϵ-nets is that if the sample S drawn by a learning algorithm forms an ϵ-net for c, and the learning algorithm outputs a hypothesis $h \in C$ that is consistent with S, then this hypothesis must have error less than ϵ. Thus if we can bound the probability that the random sample S fails to form an ϵ-net for c, then we have bounded the probability that a hypothesis consistent with S has error greater than ϵ.

For this discussion we will use a sample set that is build of two parts.
Assume the sample S_1 of m examples drown according to D, is not an ϵ-net. Denote this event as A. We want to bound the probability of this event as it bounds the probability of failure.
Assume A holds. Then there are concepts in $B_\epsilon(c)$ which are consistent with S_1. Let h be an ϵ-bad hypothesis consistent with S_1.
Designate an additional sample S_2 of m points. The expectation of the error of h is at least ϵ, thus with a probability of $\frac{1}{2}$ there will be more than $\frac{\epsilon m}{2}$ errors, for $m = |S_2| = O(\frac{1}{\epsilon})$

Define B as the event that there exists a function $h \in B_\epsilon(c)$ such that h is consistent with S_1 and has $\frac{\epsilon m}{2}$ errors on S_2. Thus:

$$Pr[B|A] \geq \frac{1}{2}$$

but

$$Pr[B] = Pr[B|A] \cdot Pr[A]$$

from which follows:

$$2 \cdot Pr[B] \geq Pr[A]$$

We can thus first find a bound on the probability of B, and this will apply a bound on the probability of A. The main advantage is that the event B is defined on the finite set of points $S_1 \cup S_2$.
Let's define F as the projection of C to $S_1 \cup S_2$. Formally:

$$F = \Pi_c(S_1 \cup S_2)$$

Later we will bound the size of F ($|F|$).

We will define the group of errors in h as follows:

$$ER(h) = \{x : x \in S_1 \cup S_2 \;\; and \;\; c(x) \neq h(x)\}$$

We assumed that $ER(h)$ has at least $\frac{\epsilon m}{2}$ elements because in S_2 there are at least $\frac{\epsilon m}{2}$ elements from $ER(h)$. That is, $|ER(h)| \geq \frac{\epsilon m}{2}$.
We are interested in the events:

$$\text{Event } A\text{: } ER(h) \cap S_1 = \emptyset$$

and

$$\text{Event } B\text{: } \begin{cases} ER(h) \cap S_1 = \emptyset \\ ER(h) \cap S_2 = ER(h) \end{cases}$$

We wish to analyze the probability that $h \in B_\epsilon$ stays consistent with S_1 and that S_2 has at least $\frac{\epsilon m}{2}$ errors. Since we chose both S_1 and S_2 from the i.i.d. distribution D, we can build the distribution on S_1 and S_2 as follows: We sample $2m$ points $S_1 \cup S_2$ and divide the sample randomly, between S_1 and S_2. This is exactly the same distribution, because any ordering of the $2m$ elements, separated into 2 groups randomly, is the same as sampling S_1 and then S_2 (due to the i.i.d property).

Our problem is now reduced to the following simple combinatorial experiment: we have $2m$ balls (the set $S = S_1 \cup S_2$), each colored black or white, with exactly $l \geq \frac{\epsilon m}{2}$ black balls (these are the points of S that h fails on them). We divide these balls randomly into two groups of equal sizes S_1 and S_2, and we are interested in bounding the probability that all the black balls fall in S_2.

Let's calculate the number of possible divisions. The number of ways we can choose l elements from $2m$ elements is:

$$\binom{2m}{l}$$

To see that this is the number of ways, assume that slots 1 to m are S_1 and $m+1$ to $2m$ are S_2. We want to place l black balls in the $2m$ slots, so this is the number of possible placements. The probability that all of the l black balls fall into S_2 is exactly

$$\frac{\binom{m}{l}}{\binom{2m}{l}}$$

Since, for all the balls to be in S_2 we have only m slots for l balls, we get:

$$\frac{\binom{m}{l}}{\binom{2m}{l}} = \prod_{i=0}^{l-1} \frac{m-i}{2m-i} \leq \frac{1}{2^l}$$

The last inequality is an approximation, assuming that each black ball can fall into S_2 with probability $\frac{1}{2}$ - thus the probability that all the black balls will fall into S_2 is $\frac{1}{2^l}$. This is of course not accurate, as the black balls' probabilities are not independent (we must have exactly m balls in each group), but this is a good approximation.

We can now bound the probabilities of A and B.

$$Pr[B] \leq |F| \cdot 2^{-l} \leq |F| \cdot 2^{-\epsilon m/2}$$

Hence:

$$Pr[A] \leq 2Pr[B] \leq 2|F| \cdot 2^{-\epsilon m/2}$$

Thus, in order for our confidence level (δ) to satisfy our goal, we will require:

$$\delta \leq Pr[A]$$

And we get that the sample size should be:

$$m = O(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{1}{\epsilon}\log|F|)$$

The only issue we still have to resolve is a bound on the size of F. As we recall, F is a projection of c on a set with $2m$ elements. We will show, that $\Pi_c(S_1 \cup S_2) \sim (2m)^d$. We will define a recursion which we will later prove bounds the number of concept in the projection.

Definition Define the function J as follows:

$$J(m,d) = j(m-1,d) + J(m-1,d-1)$$

with the initial conditions:

$$J(m,0) = 1$$

$$J(0,d) = 1$$

Solving this recursion (not detailed here) gives:

$$J(m,d) = \sum_{i=0}^{d} \left(\begin{array}{c} m \\ i \end{array} \right) \sim m^d$$

We will use this function to bound $\Pi_c(S)$ which will yield a bound on $|F|$.

Claim 7.4 *Let $VC - dim(C) = d$ and $|S| = m$, then*

$$\Pi_c(S) \leq J(m,d)$$

Proof: The proof is by induction on both d and m. For the base cases, the claim is easily established when $d = 0$ and m is arbitrary, and when $m = 0$ and d is arbitrary.
We assume for induction that for all m', d' such that $d'+m' \leq d+m$, we have $\Pi_c(S) \leq J(m,d)$. We now show that this inductive assumption establishes the desired statement for d and m. Let $S = \{x_1, \ldots, x_m\}$ be a set of m different points and let C_S be the projection of the concept class C on S. Namely,

$$\Pi_c(S) = C_S = \{c \cap S | c \in C\}$$

We will show that for every $S : |C_S| \leq J(m,d)$.
We define a new set T which is the set S after extracting the last point:

$$T = \{x_1, \ldots, x_{m-1}\} = S - \{x_m\} \quad , \quad |T| = m-1$$

Define C_* as all the assignments over T which can be completed either by $x_m = 0$ or by $x_m = 1$. Then $|C_*|$ counts the number of pairs of sets in $\Pi_C(S)$ that are collapsed to a single representative in $C_T = \Pi_C(S - \{x_m\})$. We thus have:

$$|C_S| = |C_T| + |C_*|$$

Trivially, every concept in C_S appears in C_T, and if it appears twice it is also counted in C_*.

We now bound C_T and C_* separately. The bound for C_T, from the induction hypothesis, is

$$|C_T| \leq J(m-1, d)$$

We claim that the bound for C_* is,

$$|C_*| \leq J(m-1, d-1)$$

Note that if C_* shatters a set $\{x_1, ..., x_i\}$ then C shatters the set $\{x_1, ..., x_i, x_m\}$, since each function can be completed in two different ways. By definition of C_*, for every assignment of $x_1, ..., x_i$ there exist a pair of concepts: $c_0, c_1 \in C$ that are consistent with $c_1(x_m) = 1$ and $c_0(x_m) = 0$. Hence, if C_* shatters a set of size i, then C shatters a set of size $i+1$. Since we assume $VC - dim(C) = d$, then $VCdim(C_*) \leq d - 1$.
Let's look at an example: Assume $x_1, x_7, x_{12} \in T$, are shattered by C. Clearly, they will also be shattered by C_*. But then, x_1, x_7, x_{12}, x_m are also shattered by C, as it includes both 0 and 1 as assignments for x_m. Thus, if C_* shatters k points, C shatters $k+1$ points.
Hence:

$$|C_S| = |C_T| + |C_*| \leq J(m-1, d) + J(m-1, d-1) = J(m, d)$$

which concludes the proof of the Claim. □

Let's explore the function J. As we recall,

$$J(m, d) = \sum_{i=0}^{d} \left(\begin{array}{c} m \\ i \end{array} \right) \sim m^d$$

This function has two behaviors:

$$\sum_{i=0}^{d} \left(\begin{array}{c} m \\ i \end{array} \right) \sim m^d = \left\{ \begin{array}{ll} 2^m & d \geq m \\ (2m)^d & d \ll m \end{array} \right.$$

That is, the function grows exponentially with d until d reaches m and then it grows exponentially with m. From that we can conclude that the number of functions in the projection can either grow as 2^m or fall to $(2m^d)$. No intermediate behaviors exist.
Hence:

$$\begin{aligned} & m \geq C(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{1}{\epsilon} \log m^d) \\ \Rightarrow \quad & m \geq C(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{d}{\epsilon} \log m) \\ \Rightarrow \quad & m \geq C(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{d}{\epsilon} \log \frac{d}{\epsilon}) \end{aligned}$$

As we can see above, we can actually bound the size of the sample with a function of the $VC - dim$ alone.
It is also worth noting that the difference between the lower bound and the upper bound we found are relatively small.

7.6 Bibliographic Notes

The presentation of the material of this lecture follows closely [1].

1. An introduction to Computational Learning Theory.