Machine Learning: Foundations	**Spring Semester, 1996/7**

Lecture 8: April 30, 1997

Lecturer: Yishay Mansour *Scribe: Yuval E. Sapir & Assaf K. Paznerin 1995-6* [1]

8.1 THE VC-DIMENSION

8.1.1 Motivation

In this lecture, we 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 the previous lectures we discussed the case of a finite concept class C. For that case we proved a lower bound on the number of examples required for PAC learning:

$$m \geq \frac{1}{\epsilon} \ln \frac{|C|}{\delta}$$

We also proved that in the case when the $VCdim$ is infinte, there is no static learning algorithm.

In this section we aim at handling concept classes where C is infinite, and perhaps even not enumerable. Frequently, there is a significant structure in C, and we like to formally quantify how this structure helps our learning algorithms. We will show the connection between the $VCdim$, which we defined in the previous lecture, and learning.

8.1.2 Definitions

Now define some of the entities that will be discussed. Assume C is a concept class defined over instance space X.

Definition For each class C over X and for any $\{x_1, \ldots, x_m\} = S \subseteq X$:

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

This is the projection of the concept class C on the input S, namely $\Pi_C(S)$ includes all the possible functions that C creates on S. We are interested in how many different functions C creates on S i.e. $|\Pi_C(S)|$. In effect we are reducing the concept class C into the concept

[1] Based on scribe written by Michal Hadary, Helit Heffer & Uri Blackman

class $C_{\downarrow S}$, Which is the restriction of C to S. The concept class $C_{\downarrow S}$ is finite with at most 2^m different concepts, 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 *VCdim (Vapnik-Chervonenkis dimension)* of C is the maximum size set S that is shatter 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.

Next we define for a concept class C, how many different assignments it might have on m points.

Definition

$$\Pi_C(m) = max\{|\Pi_C(S)| : |S| = m\}$$

$\Pi_C(m)$ is the maximum number of assignments over all possible samples of size m, clearly $\Pi_C(m) \leq 2^m$. If for every m we have $\Pi_C(m) = 2^m$ (i.e. for every number of points m, there exist a set of size m which C shatters) then $VCdim(C) = \infty$

8.2 Upper Bounds

8.2.1 Upper bound for VCdim(C)

Theorem 8.1 *If the concept class C is finite then $VCdim(C) \leq log_2|C|$.*

Proof: Assume $VCdim(C) = d$. So there are at least 2^d distinct concept functions in C. Therefore $|C| \geq 2^d$, from which follows: $|C| \geq 2^d$ thus $\log_2(|C|) \geq d$. □

8.2.2 A polynomial bound on $\Pi_C(S)$

Now we bound $\Pi_C(S)$ with a function $\Phi_d(m)$, where $m = |S|$ and $d = VCdim(C)$. First let us define $\Phi_d(m)$:

Definition For any natural numbers m and d,

$$\Phi_d(m) = \Phi_d(m-1) + \Phi_{d-1}(m-1)$$

with initial conditions $\Phi_0(m) = 1$ and $\Phi_d(0) = 1$.

Later we will show that

$$\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i}$$

. This will imply that $\Phi_D(m) = O(m^d)$ (later we will show the recurence exactly). The parameter d is the VC-dimension of the class and m is the number of points, which we're interested in.

First we would like to demonstrate $\Phi_d(m)$, by giving few examples:

1. As an example, consider the concept class C_1, where the concepts are c_α for $\alpha \in [0, 1]$:

   Examples:

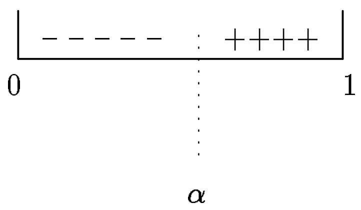

Figure 8.1: Example 1: c_α

   We showed that $VCdim(C_1) = 1$. It is easy to see that $\Pi_C(m) = m + 1$, since for each set of m points we have the folowing $(m + 1)$ assignments: for each $i \in \{0, \ldots, m\}$ we have the assignment:

$$< x_1, - >, \ldots, < x_i, - >, < x_{i+1}, + >, \ldots, < x_m, + >$$

   For this class, $d = 1$ and

$$\Phi_1(m) = \Phi_1(m-1) + \Phi_0(m) = \Phi_1(m-1) + 1 = m + 1$$

   which agrees with our computation.

2. Let us consider the concept class C_2, where the concepts are c_{α_1,α_2} for $\alpha_1, \alpha_2 \in [0, 1]$:

   Examples:

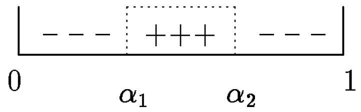

Figure 8.2: Example 2:c_{α_1,α_2}

For this example $VCdim(C_2) = 2$ as shown in the previous lecture. Folowing the same principle of the former example, $\Pi_C(m) = \binom{m+1}{2} + 1 = \binom{m}{2} + \binom{m}{1} + 1$. Again et us compute $\Phi_2(m)$:

$$\Phi_2(m) = \Phi_2(m-1) + \Phi_1 m = \Phi_2(m-1) + \Phi_1(m-1) + 1 =$$

$$= \binom{m}{2} + \binom{m}{1} + \binom{m}{0} = \binom{m}{2} + m + 1$$

The next lemma demonstrates that if $VCdim$ is finite, the number of different functions that are possible to receive by m points, $\Pi_C(m)$ grows relatively slowly, namely it is bounded by $\Phi_D(m)$, which is polynomial in m for a fixed d.

Lemma 8.2 *If $d = VCdim(C)$ then for any m,*

$$\Pi_C(m) \leq \Phi_d(m)$$

Proof: The proof is by induction on both d and m. For the base cases, the lemma 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

$$\Pi_C(m') \leq \Phi_{d'}(m')$$

We now show that this inductive assumption establishes the desired statement for d and m. We choose m different points and denote this set by $E = \{x_1, \ldots, x_m\}$. Let $C_{\downarrow E} = C_E$, the projection of the concept class C on E.

$$C_E = C_{\downarrow E} = \{c \cap E | c \in C\}$$

Namely, C_E are all the possible functions that C induces on the points $x_1 \ldots x_m$. We show that for every $E : |C_E| \leq \Phi_d(m)$. We define a new set $F = E \backslash \{x_m\}$ and $C_F = C_{\downarrow F}$, i.e. all the possible assignments on the points F. (Now we are actually looking at $(m-1)$ points and all the possible concepts). Define C_* as all the assignments over F 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 $\Pi_C(S - \{x_m\})$.We have:

$$|C_E| = |C_F| + |C_*|$$

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

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

$$|C_F| \le \Pi_{C_F}(m-1) \le \Phi_d(m-1) \ .$$

We claim that the bound for C_* is,

$$|C_*| \le \Phi_{d-1}(m-1) \ .$$

Note that if C_* shatters $\{x_1, ..., x_i\}$ then C shatters $\{x_1, ..., x_i, x_m\}$, 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 $VCdim(C) = d$, then $VCdim(C_*) \le d-1$, Hence,

$$|\Pi_C(E)| = |C_E| = |C_F| + |C_*| \le \Phi_d(m-1) + \Phi_{d-1}(m-1) = \Phi_d(m)$$

which concludes the proof of the Lemma. □

8.2.3 Analysis of $\Phi_d(m)$:

We compute the recurance of $\Phi_d(m)$ exactly in the following lemma.

Lemma 8.3

$$\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i}$$

Proof: Proof by induction on $d+m$. For $d = 0$, $m = 0$ the lemma evaluates as required. Recall that,

$$\Phi_d(m) = \Phi_d(m-1) + \Phi_{d-1}(m-1) \ .$$

For the induction step we have:

$$\Phi_d(m) = \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} = \binom{m-1}{0} + \sum_{i=1}^{d} \left[\binom{m-1}{i} + \binom{m-1}{i-1} \right]$$

We use the identity $\left(\begin{array}{c} m \\ i \end{array}\right) = \left(\begin{array}{c} m-1 \\ i \end{array}\right) + \left(\begin{array}{c} m-1 \\ i-1 \end{array}\right)$. This identity can be proved using the following argument. The choice of i elements out of m can be described by taking x_1 and another $i-1$ from the remaining $m-1$ elements, or taking i elements from the remaining $m-1$ elements, therefore :

$$\Phi_d(m) = \left(\begin{array}{c} m \\ 0 \end{array}\right) + \sum_{i=1}^{d} \left(\begin{array}{c} m \\ i \end{array}\right) = \sum_{i=0}^{d} \left(\begin{array}{c} m \\ i \end{array}\right)$$

$\square$

Using Sterling's formula $\Phi_d(m) \leq \left(\frac{2em}{d}\right)^d$ which naturally is approximately m^d.

8.3 A Polynomial bound on the sample size for PAC learning

8.3.1 ϵ-net notion

We'll show that we can use $\Phi_d(m)$ instead of the size of class $|C|$.

Definition Given a target concept C_t, we shall define all the concepts that are distanced from it by at least ϵ in relation to the distribution D. Let us define the following group:

$$\Delta_\epsilon(c_t) = \{c | Pr_D(c \neq c_t) \geq \epsilon\}$$

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_t with respect to a distribution D, if for each concept $c \in \Delta_\epsilon(c_t)$ there exists a point $x_i \in S$ such that $c(x_i) \neq c_t(x_i)$.

The important property of ϵ-nets is that if the sample S drawn by a learning algorithm forms an ϵ-net for c_t, 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_t, then we have bounded the probability that a hypothesis consistent with S has error greater than ϵ.

Let us show an example of ϵ-net:

reconsider the concept class C_1.

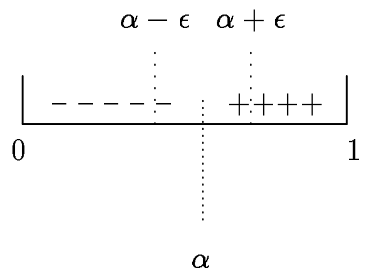

Figure 8.3: Example of an ϵ-net

Each set that includes $x_1 \in (\alpha-\epsilon, \alpha)$ and $x_2 \in (\alpha, \alpha+\epsilon)$ constitutes an ϵ-net with respect to the uniform distribution over $[0, 1]$.

8.3.2 A small ϵ-net from random sampling

We now show that if we draw a small set of examples from the oracle $EX(c, D)$, then they form an ϵ -net with high probability. The important property is that the size of the required sample depends on the $VCdim$ (d) and ϵ and δ, but is independent of $|C|$ and $|X|$. From the preceding discussion, this will immediately lead to an upper bound on the number of examples required for PAC learning that depends only on these same quantities.

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. If A occured, then there are concepts $r \in \Delta_\epsilon(c_t)$ which are consistent with S_1. Designate an additional sample S_2 of m points. On average, the error of r is at least ϵ, thus with a probability of $\frac{1}{2}$ there will be more than $\frac{\epsilon m}{2}$ errors, for $m = O(\frac{1}{\epsilon})$

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

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

but

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

from which follows:

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

First we'll find a bound on the probability of B, and this applies a bound on the probability of A. The main advantage is that the event B is defined on the finite set of points $S_1 \cup S_2$. In addition, we will later use the following fact:

$$\Pi_{\Delta_\epsilon(C)}(S_1 \cup S_2) \leq \Phi_d(2m)$$

We first analyze the probability that $c \in \Delta_\epsilon$ stays consistent with S_1 and that S_2 has at least $\frac{\epsilon m}{2}$ errors. We choose both S_1 and S_2 from the distribution D. The distribution is i.i.d. (identically independently distributed), thus 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, seperated into 2 groups randomly, is the same as sampling S_1 and then S_2 (due to the i.i.d property).

Definition

$$E_c = \{x : x \in S_1 \cup S_2 \ \ and \ \ c(x) \neq c_t(x)\}$$

We assumed that E_c has at least $\frac{\epsilon m}{2}$ elements because in S_2 there are at least $\frac{\epsilon m}{2}$ elements from E_c. We are interested in the event:

$$S_1 \cap E_c = \emptyset \ \ and \ \ S_2 \cap E_c = E_c.$$

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 c fails on them). We divide these balls randomly into two groups of equal size S_1 and S_2, and we are interested in bounding the probability that all l of the black balls fall in S_2.

The number of possible divisions: From $2m$ elements we can choose l elements in the following number of ways,

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

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 placements The probability that all l of the black balls fall on 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. Since,

$$\frac{\binom{m}{l}}{\binom{2m}{l}} = \prod_{i=0}^{l-1} \frac{m-i}{2m-i} \leq \prod_{i=0}^{l-1} \left(\frac{1}{2}\right) = \frac{1}{2^l} \leq \frac{1}{2^{\frac{\epsilon m}{2}}}$$

Where in the last inequality we used the fact the $l \geq \frac{\epsilon m}{2}$. We get the following bound for $\Pr[B]$:

$$Pr[B] \leq |\Pi_{\Delta_\epsilon(c_t)}(S_1 \cup S_2)| \cdot 2^{-\frac{\epsilon m}{2}} \leq \Phi_d(2m) \cdot 2^{-\frac{\epsilon m}{2}} \leq \left(\frac{2em}{d}\right)^d \cdot 2^{-\frac{\epsilon m}{2}}$$

Finally,

$$Pr[A] \leq 2Pr[B] \leq 2\left(\frac{2em}{d}\right)^d \cdot 2^{-\frac{\epsilon m}{2}}$$

which is less than δ for

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

Theorem 8.4 *Let C be a concept class with $VCdim(C) = d$. Let A be any algorithm that takes as input a set S of m labeled examples of a concept in $C_t \in C$, and produces as output a concept $h \in C$ that is consistent with S. Then A is a PAC learning algorithm for C provided it is given a random sample of m examples from $EX(c, D)$, where m obeys*

$$m \geq c_0(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{d}{\epsilon}\log\frac{1}{\epsilon})$$

for some constant $c_0 > 0$.

8.4 Learning Neural Networks

8.4.1 The basic unit of Neural Networks

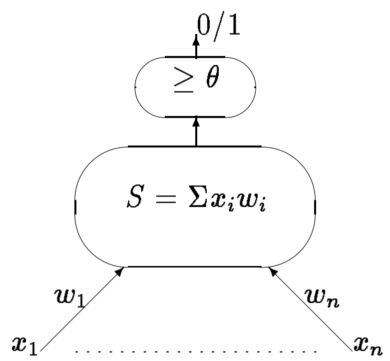

Figure 8.4: The basic unit of neural network

Each basic unit has n inputs $(x_1, \ldots, x_n)$. When $\Sigma x_i w_i \geq \theta$ the unit outputs one. Otherwize, it outputs zero.[2]Note that $w_1, \ldots, w_n$ and θ are the parameters of the unit, i.e. they are constant for any set of inputs.

8.4.2 The structure of a Neural Network

Given these basic units, we can combine them to create Neural Networks (NN) much like combining and/or gates creates boolean circuit. The NN is an acyclic directed graph for which the output of an edge is marked by an outgoing vetrex, and the input by ingoing one. Each vertex in layer $l+1$ receives its inputs from layer l.

The NN has three basic parameters:

- The number of inputs denoted n.
- The number of nodes denoted s.
- The in degree of a non-input node denoted r.

8.4.3 An upper bound for Neural Networks

We consider a general case in which we have a family of concepts C over $\Re^r$ to $\{0,1\}$,and each node has a concept $c \in C$. Since each node can output only zero or one, the input values of all inner nodes (including the root) will be from $\{0,1\}^r$.

The object of learning the NN is finding which concept resides in each node, i.e. the structure of the NN is known to the learning algorithm.

We shall denote the family of functions produced by assigning concepts from C to nodes of the graph G, by C_G.

The objective of the following theorem is to bound $VCdim(C_G)$ by a polynomial function of $VCdim(C)$.

Theorem 8.5 *Let G be an acyclic directed layered graph with n input nodes (zero entry level), $s \geq 2$ inner nodes with an entry level of r. Let C be a concept class over $\Re^r$ with $VCdim(C) = d$. Then*

$$VCdim(C_G) \leq 2ds \cdot log(es)$$

Proof: The idea is to bound $\Pi_{C_G}(m)$. Once we have an upper bound on it, then the first value of m for which $\Pi_{C_G}(m) < 2^d$ in necessarily an upper bound on $VCdim(C_G)$.

Let $S = \{\vec{x_1}, \ldots, \vec{x_m}\}$, where $\vec{x_i} \in \Re^n$. Each vector $\vec{x_i}$ is an input to the NN (the graph). Once we set the concepts for all the nodes in G, we can compute $G(\vec{x_i})$.

[2]Usually it is preffered that the output of the unit will be continuous and differentiable, for example, outputing $\frac{e^s}{1+e^s}$ where $s = \Sigma x_i w_i$, but here we analyze the simple case of a sharp threshold.

For each node in G we build a vector from $\{0,1\}^m$ by setting the i^{th} bit to be the note's output when $\vec{x_i} \in S$ is the input of the graph. i.e. this vector describes the outputs of a node with respect to each vector in S.

Cosider all the possible functions in C_G, i.e., all assignments of concepts from C to all nodes of G. Two functions are distinct with respect to S, then there exists at least one node in G, for which the above mentioned vectors differ.

Let $T_{C_G}(S)$ denote the number of different computations of functions from C_G with S as inputs. In other words, this is the number of distinct functions in C_G (as described above).

Clearly $|\Pi_{C_G}(S)| \leq T_{C_G}(S)$, since $\Pi_{C_G}(S)$ considers two functions to be distinct only when the output vector (the vector of the root) is different, without considering all the vectors of the other nodes.

In order to bound $T_{C_G}(S)$ we extract the root of graph G (denoting the new graph G'). Let $T_{C_{G'}}(S)$ be the number of different computations of G' over S. Each computation can be extended into at most $\Pi_C(m)$ different computations of G over S (For each computation of G' the root gets $y_1, \ldots, y_r$ vectors of size m, which construct up to m different inputs. Assigning different concepts to the root can expand these inputs by magnitude of $\Pi_C(m)$.) Therefore

$$|\Pi_{C_G}(S)| \leq T_{C_G}(S) \leq T_{C_{G'}}(S) \cdot \Pi_C(m)$$

By continuing the process of root extracting and we have

$$|\Pi_{C_G}(S)| \leq |\Pi_C(M)|^s \leq \left(\frac{em}{d}\right)^{ds}$$

where the second inequality is using Sterling's formula. Note that we can use the depth of the NN instead of the number of nodes, s.

Our goal was to find m such that $\Pi_{C_G}(m) < 2^m$, and this m would bound the $VCdim(C_G)$. We can achieve this by finding an m for which

$$\left(\frac{em}{d}\right)^{ds} < 2^m.$$

Any m that satisfies,

$$m > 2ds \cdot log_2(es),$$

with the condition that $s \geq 2$ will do. □

We now use this theorem in order to bound the $VCdim$ of NN when $C = \{\sum_{i=1}^{r} w_i x_i \geq \theta\}$. Note that this concept class is actually the class of all hyper-planes over $\Re^r$, for which we already know that $VCdim(C) = r + 1$[3]. Therefore

$$VCdim(C_G) \leq 2(r+1)s \cdot log(es)$$

Combining this conclusion with the upper bound on the sample size needed for PAC learning we have

[3]See previous lecture

Theorem 8.6 *Let G be a acyclic directed graph, C_G the collection of all possible NN over G with s inner nodes and an in degree of r. The number of samples sufficient for PAC learning of C_G is*

$$O\left(\frac{1}{\epsilon}log\frac{1}{\delta}+\frac{(r+1)s\cdot log(s)}{\epsilon}log\frac{1}{\epsilon}\right)$$

The lecture follows closely chapter 3 in Kearns and Vazirani.