A seminar on learning theory

Instructed by Amir Yehudayoff, Department of Mathematics, Technion-IIT

Contents

Chapter 0

Outline

The aim this seminar is to survey, discuss and explore mathematical aspects of learning. As we shall see, learning is related to questions in geometry, topology, combinatorics, computational complexity, logic, and more.

The participants are expected to take an active part in the meetings, and to present some work. From time to time, exercises will be given to help digest the material, and a final assignment may be given as well.

Chapter 1

Introduction

"Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense."

A theory of the learnable / L. Valiant

Examples. Before providing formal definitions, let us give some examples.

- Sequences: $0, 1, 1, 2, 3, 5$ what is next?
- Shapes: draw some points with yes/no labels. What is shape?
- Image recognition.
- Your examples?

The psychological aspects of learning are fascinating, but we focus will be on the mathematics of it. For this we need definitions.

How do we abstractly define the concept we aim to learn? All examples above can be abstracted by a single object, a function: $\mathbb{N} \to \mathbb{N}$, $\mathbb{R}^2 \to \{0, 1\}$, and say $[1000]^2 \to \{a, b, \ldots, z\}^{10}$.

To what space does the concept belong to? We can try the space of all functions say $\mathbb{N} \to \mathbb{N}$ in the first example. In this case, how do we continue the sequence? There is no unique extension. But intuitively, or from experience, we know that the next element is 8. We model this by restricting the space of functions. Roughly speaking, we reduce it to the space of functions with "simple" explanation or description. In the example, we can reduce it to the vector space of sequence that are Fibonacci and

then the next element is uniquely defined. More abstractly, we choose some combinatorial/geometric/topological ways to restrict the function space.

Definition 1. *The instance space is a set X. The label set is a set Σ. A concept is $c : X \to \Sigma$. A concept class is $C \subseteq \Sigma^X$.*

In many cases we are interested in classification $L = \{0, 1\}$. We think of C and X as known. There is some $c \in C$ that is unknown and we wish to learn. How do we define "learn"? There are many ways, as we shall see. Can you provide some suggestions?

What does the learner see? The learner gets to see some information concerning c. The simplest type is samples of the form $(x_1, c(x_1)), (x_2, c(x_2)), \ldots$ and from this info he needs the recover c.

Consider the example of $X = \mathbb{R}$ and C the set of polynomials of degree at most d. In this case, from $d+1$ labelled samples we can fully reconstruct c as long as the instances $x_1, \ldots, x_{d+1}$ are pairwise distinct. This is done by interpolation: $c = \sum_{j=0}^{d} a_j x^j$ and

$$c(x_i) = \sum_{j=0}^{d} a_j x_i^j.$$

Stated differently the evaluation vector $e = (c(x_i))$ is obtained by multiplying a $d+1 \times d+1$ Vandermonde matrix A by the vector of coefficients a of c. That is, $e = Aa$. So $a = A^{-1}e$ gives us a full description of c.

What if we do not get $d+1$ distinct evaluation points? Then we can not in general uniquely recover c. So part of the definition should specify the way samples are gathered (cleverly, randomly, adversarially,...).

1.1 Several models

Let us consider another example. Let $X = [n] \subset \mathbb{R}$. Let $C : X \to \{0, 1\}$ be the set of c_i, $i \in [n]$, so that $c_i(x) = 1$ iff $x \geq i$. Roughly speaking, it is a family of n rays. We discuss several learning frameworks, using this example.

Self directed learning. The student chooses the sample points according to an order of his choice $x_1, x_2, \ldots, x_n$ in an adaptive way. At every stage t the student chooses a point x_t that he has not seen so far. He also chooses an hypothesis $h_t : X \to \Sigma$ that may depend on what he has seen so far, and is assumed consistent with what he has seen so far. (We may insist that h_t is in C, which is called proper learning.) He then declares $c(x_t) = h_t(x_t)$. If he is wrong then it costs him a point, and if he is correct then he just keeps learning. He then chooses the next hypothesis h_{t+1} and the next point x_{t+1}.

Let us give a formal definition. Fix a concept class C. For every adaptive order $\pi = \{x_1, x_2, \ldots, x_n\}$ on X, and for every choice $h = (h_1, \ldots, h_n)$ of n functions as above, the cost of learning $c \in C$ is defined is

$$cost_c(\pi, h) = |\{t \in [n] : h_t(x_t) \neq c(x_t)\}|.$$

The self directed learning complexity of c in C is defined as

$$SDL(C) = \min_{\pi} \min_{h} \max_{c \in C} cost_c(\pi, h).$$

It is the minimum over all student, and for each student the maximum over all concepts.

What should the student/learner do to minimize the cost of learning in our running example? Here is a suggestion. He chooses $x_t = t$. He declares $h_1(1) = 0$. If he is wrong, $c(1) = 1$ which means that $c = c_1$. If he is right then there is no cost, and he continues with $h_2(2) = 0$. If he is wrong, $c = c_2$, and otherwise he continues. Overall, the cost of this learner is just 1 point. Other students may make many more mistakes.

Online learning (online mistake bound). In this model, the order on X is chosen by an adversary. The same cost is defined. Roughly speaking, here the learner is measured against the worst or most challenging teacher, rather than his own choices. The complexity is

$$OL(C) = \max_{\pi} \min_{h} \max_{c \in C} cost_c(\pi, h).$$

Clearly, $OL(C) \geq SDL(C)$.

What is the worst order in running example? Choose $x_1 = n/2$. If the student says $h_1(x_1) = 0$, then he "goes right" and otherwise we go left.

Exercise: Show that the online complexity of C from the running example is $\lceil \log n \rceil$.

Probably approximately correct (PAC) learning. This model was defined by Valiant, and it was a seminal definition.

In this model there is a probability distribution μ on X. (We assume X is finite for now.) The learner gets to see random examples of the form $(x, c(x))$ where $x \sim \mu$. The distribution μ is meant to capture the frequency according to which examples appear in nature. The learner does not know μ.

The learner objective is: given as few examples as possible $(x_1, c(x_1)), \ldots, (x_m, c(x_m))$, where $x_1, \ldots, x_m$ are i.i.d. according to μ, output an hypothesis $h : X \to \Sigma$ that is a good approximation to c:

$$\mu(\{x : h(x) \neq c(x)\}) \leq \epsilon.$$

The learning process is random, so this should happen with probability at least $1 - \delta$.

Here is a brief geometric perspective (we shall elaborate further on in future). We think of Σ^X as a (pseudo) metric space. The distance between c, c' is

$$\mu(\{x : h(x) \neq c(x)\}).$$

PAC learning can thus be thought of as finding an approximation to a given point in the metric space using a few examples as possible.

Think of a baby seeing random pictures of objects. The baby is told how to classify each picture. His goal is to come up with a procedure to classify new picture that he has not yet seen.

Definition 2 (PAC learnability)**.** *We say that C is PAC learnable with m examples if there is map H so that the following holds. Define the set of C-labelled samples of size smaller than $1 \leq k \leq \infty$ as*

$$L_C(k) = \{(Y, y) : Y \subseteq X,\ |Y| < k,\ y \in C|_Y\}.$$

The map H generates hypotheses: $H : L_C(d+1) \to \Sigma^X$ so that for every $c \in C$ and for every probability distribution μ on X,

$$\Pr_{\mu^d}\left[\left\{Y \in X^d : \mu(\{x \in X : h_Y(x) \neq c(x)\}) \leq \frac{1}{3}\right\}\right] \geq \frac{2}{3},$$

where $h_Y = H(Y, c|_Y)$.

Roughly speaking, an hypothesis generated by H using d independent samples is a μ-approximation of c with reasonable probability. If the image of H is contained in C, we say that C is properly PAC learnable.

Let us see how to PAC learn the running example. Given m examples, order them as $x_1 \leq x_2 \leq \ldots \leq x_m$ in $[n]$. If c on all of them is 0, output $h = c_{x_m+1}$. Otherwise, let j be the smallest integer so that $c(x_j) = 1$ and output $h = c_{x_j}$.

By the structure of C, we know $h \leq c$ pointwise.

Claim 3. *If $m \geq 10\log(1/\delta)/\epsilon$ then $\Pr_{\mu^m}[\mu(\{x : h(x) \neq c(x)\}) > \epsilon] < \delta$.*

Later, we shall see a more general way to prove this claim.

Proof sketch. Assume $c = c_i$.

If $\mu(i) \geq \epsilon$ then the probability that $i \notin \{x_1, \ldots, x_m\}$ is at most $(1-\epsilon)^m < \delta$. The proof is complete in this case, since if $i \in \{x_1, \ldots, x_m\}$ then $h = c$.

Otherwise, let $i' \in [n]$ be the largest integer so that $i' \geq i$ and $\mu(\{i, \ldots, i'\}) < \epsilon$.

If $i' = n$ then c is ϵ-close to the zero function, so c is also ϵ-close to any h that is at most c.

Otherwise, ...

We can keep analyzing the construction case by case, looking at properties of μ with respect to c. We will not provide full details here, since we shall prove a more general theorem later on (so called double sampling argument).

□

Chapter 2

VC dimension

In this part, we define and investigate a combinatorial property that captures PAC learnability for boolean concept classes. It was defined by Vapnik and Chervoninkis in 1971, in the context of statistics and probability theory.

Definition 4 (VC dimension)**.** *Let $C \subseteq \{0,1\}^X$. A set $Y \subseteq X$ is shattered in C if for every $Z \subseteq Y$ there is $c \in C$ so that $c|_Z = 1$ and $C|_{Y-Z} = 0$. The VC dimension of C is the maximum size of a set that is shattered in C.*

Thinking of C as a binary matrix with rows labelled by $c \in C$ and columns by $x \in X$, a subset of columns Y is shattered if all the $2^{|Y|}$ possible different zero-one patterns appear in the columns in Y. Another perspective of $VC(C)$ is the ability of concepts in C to separate points in X. The following examples may help to clarify.

Examples.

1. What are the shattered subsets of the following? What is its VC dimension?

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

2. $X = \mathbb{R}$ and C consists of all intervals.

   For every $x \in X$, there is $c_0, c_1 \in C$ so that $c_0(x) = 0$ and $c_1(x) = 1$ so $VC(C) \geq 1$.

   For every distinct $x, x' \in X$, there is $c_{00}, c_{10}, c_{01}, c_{11} \in C$ so that $c_{bb'}(x, x') = bb'$. Draw the 4 intervals. So $VC(C) \geq 2$.

   However, if $x < x' < x''$ then every interval that contains x, x'' also contains x' so that pattern 101 does not appear. Overall $VC(C) = 2$.

3. $X = \mathbb{R}^2$ and C consists of axis parallel rectangles.

   There are 4 points with 16 patterns (every subset of them can be separated).

   However, if $p_1, \ldots, p_5$ are distinct points in the plane and p_1 an uppermost point, p_2 is a lowermost point, p_3 is a leftmost point and p_4 is a rightmost point, then the pattern 11110 does not appear. So $VC(C) = 4$.

4. Union of a finite number of intervals with rational endpoints on the real line:

$$C = \{\cup_n (p_i, q_i) \mid p_i, q_i \in \mathbb{Q}, n \in \mathbb{N}\}.$$

   For any finite sample of points on the line, one can place a sufficient number of intervals to realize any labeling, therefore $VCdim(C) = \infty$.

5. Sinus:

$$C = \{sign(\sin(\omega x + \theta) : \omega, \theta \in \mathbb{R}\}.$$

   Then, $VCdim(C) = \infty$, as for every finite set of points all possible labellings can be realized by choosing a sufficiently large frequency ω and appropriate phase.

VC dimension implies PAC learnability. A fundamental and well-known result of Vapnik and Chervonenkis and of Blumer, Eherenfeucht, Haussler, and Warmuth states that every boolean concept class C can be properly PAC learned with finitely many examples iff $VC(C) < \infty$. We now state the theorem (in the harder direction), but will prove it only later on. The proof uses a clever double sampling argument.

Theorem 5. *Let $C \subseteq \{0,1\}^X$ with[1] $|X| < \infty$ and $VC(C) = d$. Let $c \in C$. Let μ be a probability distribution on X. Let Y be a multi-set of m independent samples from μ. Let h be any function in C so that $h|_Y = c|_Y$. Then, for every $\epsilon > 0$,*

$$\Pr[\mu(\{x : h(x) \neq c(c)\}) > \epsilon] \leq 2 \left(\frac{em}{d}\right)^d (1 - \epsilon/4)^m.$$

Specifically, if $\epsilon = \delta = 1/3$ then C can be PAC learnt with[2] $O(d \log d)$ many examples, with accuracy ϵ and error probability δ. The learning algorithm chooses any hypothesis that is consistent with the given samples. Note that it does not guarantee efficient learning. The proof will be given later in Chapter 3.

Some simple properties. The following operations do not increase the VC dimension.

[1] We assume this here to eliminate measurability issues.

[2] Big O and Ω notation means up to absolute constants.

1. Projections: If $Y \subseteq X$ then $VC(C|_Y) \leq VC(C)$.

2. Deletion: $VC(C - \{c\}) \leq VC(C)$.

3. Translations: For every $v \in \{0,1\}^X$, we have $VC(v+C) = VC(C)$ where addition is modulo 2.

Exercise: What is the VC dimension of the following classes?

- A linear subspace of dimension k in $\mathbb{F}_2^n$.
- The set of all convex shapes in the plane.
- The set of triangles in the plane.

Halfspaces An important concept class is that of halfspaces. Let $X \subseteq \mathbb{R}^d$ and $C \subseteq \{0,1\}^X$ be the defined by halfspaces: $c \in C$ iff there are $\alpha \in \mathbb{R}^d$ and $\theta \in \mathbb{R}$ so that $c(x) = 1$ iff $\sum_i \alpha_i x_i \geq \theta$. Draw in the plane.

This type of functions appear in many places: passing a course, economics, some models of the brain, ...

Claim 6. *If C is the set of halfspaces in the plane then $VC(C) = 3$.*

Sketch. Every 3 points that are not collinear can be separated by C.

If we choose 4 points in the plane, there are 2 options. One is that they are not convexly independent, in which case the pattern 1110 does not occur. The other is that they are convexly independent, in which case the pattern 1010 does not occur. □

Exercise: What is VC dimension of halfspaces in $\mathbb{R}^d$?

2.1 Sauer-Perles-Shelah

Scribe: Shay Moran

One of the most important properties of classes of low VC dimension is that they have small projections.

Lemma 7 (Sauer-Perles-Shelah[3]). *Let $C \subseteq \{0,1\}^X$ with $|X| = n$ and $VC(C) = d$. Then,*

$$|C| \leq \binom{n}{\leq d} := \sum_{i=0}^{d} \binom{n}{i} \leq \left(\frac{en}{d}\right)^d.$$

We shall see its importance later on and also prove in 2 ways. We first show the second inequality: By the binomial identity and the fact that $(1+x) \leq e^x$,

$$\left(\frac{d}{n}\right)^d \sum_{i=0}^{d} \binom{n}{i} \leq \sum_{i=0}^{d} \left(\frac{d}{n}\right)^i \binom{n}{i} \leq \sum_{i=0}^{n} \left(\frac{d}{n}\right)^i \binom{n}{i} = \left(1+\frac{d}{n}\right)^n \leq e^d.$$

Tightness. Consider the hamming ball of radius d around 0. How many concepts are there? What is the VC dimension? The hamming ball demonstrates the tightness of Sauer's lemma. Concept classes for which Sauer's lemma is tight are called maximum classes. They have a rich combinatorial structure, and were studied in combinatorics, geometry and machine learning.

2.1.1 First proof: Shifting

The first proof uses an operation that does not increase the VC dimension but leaves the size as is: Shifting. This is an important technique and has several other variants than we present here (and more applications).

High level. Let $C \subseteq \{0,1\}^n$ be a concept class of VC dimension d. We will transform C to a concept class C_{end} with the following properties:

- $|C_{end}| = |C|$, and
- C_{end} is a subset of a hamming ball of radius d.

Sauer's lemma then follows (why?).

Notation. Let $c_1, c_2 \in \{0,1\}^n$, and let $x \in [n]$. We say that c_1 and c_2 are x-neighbors if c_1 and c_2 agree on $[n] - \{x\}$ and disagree on x. We say that C is downward closed if for every $c_2 \in C$, and $c_1 \leq c_2$, it holds that $c_1 \in C$.

Example. The running example is not downward closed, and any hamming ball around the 0 concept is downward closed.

[3]Some people call it the Sauer-Perles-Shelah-Perles lemma, since Perles proved it twice, independently.

Shifting. For $x \in [n]$, define a shifting operator S_x in the x'th direction. The class $S_x(C)$ is a (new) concept class of the same size as C that is obtain by pushing the concepts in C downwards in the x'th direction if possible. That is, $S_x(C)$ is obtained from C by the following:

For all $c \in C$ such that $c(x) = 1$, if the x-neighbor of c is not in C then replace c by its x-neighbor.

Example. Shifting the running example once in each coordinate. What are the concept classes for which $S_x(C) = C$ for all $x \in [n]$?

Claim 8. *$VC(S_x(C)) \leq VC(C)$. In fact, if $Y \subseteq X$ is shattered in $S_x(C)$ then it is shattered in C.*

Proof. Let Y be shattered in $S_x(C)$. If $x \notin Y$ then the claim holds, since we did not change the $C|_Y$. Otherwise, every pattern that has a 1 at x in $S_x(C)$ also appears in C, since the same vector is also in C. So, consider a pattern v that has a 0 in x. Since Y is shattered by $S_x(C)$ then also v', the x-neigbour of v appears in $S_x(C)$. Therefore v' appears also in C. Now, since v' appears in both C, and $S_x(C)$, it follows that its x-neighbour, v, appears in C. So, Y is also shattered by C. □

By repeatedly shifting C, we get to a class C_{end} that is downward closed (why?). Moreover, $VC(C_{end}) \leq VC(C)$. This implies that C_{end} is contained in a hamming ball of radius d around 0. The Sauer-Perles-Shelah lemma follows.

Exercise. Let $C \subseteq \{0,1\}^n$, and let C' be a class which is obtained by shifting C once on each coordinate (in any order). Show that C' is downward closed.

In fact, the proof yields a stronger statement. Denote by $St(C)$ the set of $Y \subseteq X$ that are shattered in C. Then,

$$|C| \leq |St(C)|.$$

This is indeed stronger since

$$St(C) \subseteq \binom{X}{\leq VC(C)} := \{Y \subseteq X : |Y| \leq VC(C)\}.$$

2.1.2 Second proof: Algebraic method

The second proof is an example of using algebraic objects to prove combinatorial statements. This is a very powerful method, and has many applications. In many cases, the proof is based on judiciously defining a vector space V and computing its dimension in 2 different ways.

High level. We define a vector space V of dimension $|C|$, and prove that $dim(V) \leq \binom{n}{\leq d}$ by finding a spanning set of this size. The Sauer-Perles-Shelah lemma follows.

The vector space. Fix some field $\mathbb{F}$, say, $\mathbb{F} = \mathbb{R}$. The vector space V is the space of all functions from C to $\mathbb{F}$:

$$V = \{f : C \to \mathbb{F}\}.$$

It is indeed a $|C|$-dimensional vector space.

An equivalence relation. Let U be the space of n-variate polynomials[4] $P(x_1, \ldots, x_n)$ from $\{0,1\}^n$ to $\mathbb{F}$. Every $P \in U$ corresponds to $P|_C \in V$. But every vector in V corresponds to many elements of U. For $P, Q \in U$, write $P \sim Q$ if $P|_C = Q_C$. If $P = Q$ then $P \sim Q$ but the other direction does not necessarily holds. If C is our running example then the polynomial $P = x_1(1 - x_2)$ satisfies $P \sim 0$ but $P \neq 0$. We can also define V as $U/\sim$.

A basis. Let $U(r) \subseteq U$ be the set of all multilinear monomials of total degree at most d. Thus, $|U(r)| \leq \binom{n}{\leq r}$. Let $V(r) = U(r)/\sim$. Thus, $|V(r)| \leq |U(r)|$. The following claim thus completes the proof.

Claim 9. *The set $V(d)$ spans V.*

Proof. First, the set $U(n)$ spans U, and so $V(n)$ spans V. This is because every indicator function can be represented as a multilinear polynomial; for example, for the zero element in $\{0,1\}^n$ it is

$$\prod_{i \in [n]} (1 - x_i)$$

It remains to show that every monomial in $V(n)$ can be expressed as a linear combination of the monomials in $V(d)$. Let $x_Y = \prod_{i \in Y} x_i \in V(n)$. We proceed by induction on $|Y|$. We can of course assume that $|Y| > d$. This implies that there exist $r \in \{0,1\}^Y$ such that for all $c \in C$,

$$c|_Y \neq r.$$

Assume for simplicity that $r = 0$. Consider

$$P = \prod_{i \in Y} (x_i - 1).$$

It holds that

$$P|_C = 0.$$

[4]We use variables $x_1, x_2, \ldots$ even though x also denotes an element of $[n]$.

Specifically,

$$0 \sim P = x_Y + Q_Y,$$

where the total degree of Q_Y is smaller than $|Y|$. By induction, Q_Y is in the span of $V(d)$. Since $x_Y \sim -Q_Y$ we get that x_Y is in this span as well. □

2.2 Cover numbers

A useful perspective that we shall discuss several time throughout the seminar is that of C as a metric space. Define the normalised hamming distance on $\{0,1\}^n$ by

$$dist(c_1, c_2) = \frac{|\{i \in [n] : c_1(i) \neq c_2(i)\}|}{n}.$$

It is a metric and $0 \leq dist \leq 1$.

Theorem 10. *Let $\epsilon \in (0,1]$ such that for all $c_1 \neq c_2$ in $C \subseteq \{0,1\}^n$,*

$$dist(c_1, c_2) \geq \epsilon.$$

If $VC(C) = d$ then

$$|C| \leq \left(\frac{100 \log(2/\epsilon)}{\epsilon}\right)^d.$$

Later on, we will see a more complicated proof of a stronger statement [Haussler] saying that in fact

$$|C| \leq (100/\epsilon)^d.$$

Two comments that may help to understand the meaning of the theorem (the proofs are left as exercises):

1. If there is no constraint on the VC dimension then the size of such C can be exponentially large: There exists $C \subseteq \{0,1\}^n$ of size $|C| \geq 2^{n/100}$ such that for all $c_1 \neq c_2$ in C,

   $$dist(c_1, c_2) \geq 1/4.$$

   (Hint: a random choice of C will do.)

2. This demonstrates a similarity between VC dimension and euclidean dimension: Let $C \subseteq \mathbb{R}^n$ be such that C is contained in a linear subspace of dimension d. Assume that $\|c\| \leq 1$ for all $c \in C$, where $\|\cdot\|$ is say the L^2-norm. Let $\epsilon \in (0,1]$ such that for all $c_1 \neq c_2$ in C,

   $$\|c_1 - c_2\| \geq \epsilon.$$

Then,

$$|C| \leq (4/\epsilon)^d .$$

(Hint: a volume argument.)

The proof we present demonstrates the probabiitemizeic method, which is another central tool introduced by Erdos.

Proof. Let m denote $|C|$, and let $k = 2\log(m)/\epsilon$. Pick $x_1, \ldots, x_k$ independent uniform samples from $[n]$, and consider the random class $C' = C|_{\{x_1,\ldots x_k\}}$. Thus, $|C'| \leq |C|$. Let E denote the event $|C'| < |C|$. We shall first prove that $\Pr[E] < 1$. Indeed, for each $1 \leq i < j \leq m$, let $E_{i,j}$ be the event that c_i and c_j agree on $\{x_1, \ldots, x_k\}$. Thus,

$$E = \bigcup_{1 \leq i < j \leq n} E_{i,j}.$$

Since the distance of every pair $c_i \neq c_j$ in C is at least ϵ,

$$\Pr[E_{i,j}] \leq (1-\epsilon)^k.$$

The union bound implies

$$\Pr(E) \leq m^2(1-\epsilon)^k < 1,$$

by choice of k.

Since $\Pr[E] < 1$, there are $x_1, \ldots, x_k$ so that the size of $C' = C|_{\{x_1,\ldots,x_k\}}$ is the same as $|C|$. Now, by Sauer's lemma,

$$m = |C| = |C'| \leq \left(\frac{ek}{d}\right)^d = \left(\frac{2e\log(m)}{\epsilon \cdot d}\right)^d .$$

So,

$$m \leq \left(\frac{2e\log(m)}{\epsilon \cdot d}\right)^d .$$

This implies that (exercise)

$$m \leq \left(\frac{100\log(2/\epsilon)}{\epsilon}\right)^d .$$

□

Chapter 3

PAC learning VC classes

Scribe: Anat Kira and Yoni Mirzae

In many learning situations, the learner (or algorithm) is given m examples sampled from some fixed unknown distribution μ, and the learner generates an hypothesis based on these examples with the required accuracy and confidence.

Consider, for example, the case of classifying emails as spam/non-spam. We would like our learning algorithm to be able—after seeing m emails sampled i.i.d. and their classification—to tell if any other random email is spam or not and to be right with some required probability. The probably approximately correct (PAC) model, introduced in previous chapters, formalizes this [Valiant 84].

Our learners here will be consistent:

Definition 11. *A function H from sample space to hypothesis space is* **consistent** *if for all c, Y,*

$$H(Y, c|_Y)|_Y = c|_Y.$$

Example. The problem of learning axis-parallel rectangles in the plane is PAC-learnable. Here is an algorithm to learn a concept $c \in C$: Keep track of the minimum and maximum x and y coordinates of all positive examples. Let l', r', b', t' be these values (l for left, r for right, b for bottom and t for top). Predict that the concept is

$$h = [l', r'] \times [b', t'].$$

If there are no positive examples, let $h = \emptyset$.

Claim 12. *This algorithm is a learning function for C with sample complexity at most $\frac{4}{\epsilon}\ln(4/\delta)$.*

Proof. Assume the concept to be learned is $[l, r] \times [b, t]$. A property that always holds by construction is that the output hypothesis h satisfies $h \subseteq c$. Thus, if $P(c) < \epsilon$ then $err(h) = \mu\{h \neq c\} < \epsilon$ always and we are done. Otherwise, define 4 side rectangles within c:

$$R_{left} = [l, x] \times [b, t],$$

where

$$x = \inf\{x : \mu([l, x] \times [b, t] \geq \epsilon/4\},$$

and R_{right}, R_{bottom} and R_{top} are defined similarly. Let $Y = (x_1, ..., x_m)$ be the m input examples. Then,

$$\mu(R_{left} \textit{ contains no example }) = \mu(R_{left} \cap Y = \emptyset) \leq (1 - \epsilon/4)^m.$$

And the same holds for the other side rectangles. Let E be the event that some side rectangle contains no examples, i.e.

$$E = \{R_{left} \cap Y = \emptyset\} \cup \{R_{right} \cap Y = \emptyset\} \cup \{R_{bottom} \cap Y = \emptyset\} \cup \{R_{top} \cap Y = \emptyset\}.$$

Thus,

$$P(E) \leq 4(1 - \epsilon/4)^m \leq 4e^{-m\epsilon/4} < \delta.$$

By choice, every h the comes from samples that are not in E satisfies $dist_\mu(h, c) \leq \epsilon$, as needed. □

3.1 VC dimension

Having defined *PAC learning*, a natural question to ask is when a concept class is PAC-learnable. In order to answer this question we introduce a combinatorial measure originally defined by Vapnik and Chervonenkis (1971), thus referred to as VC dimension. The VC dimension measures the complexity, or richness, of a concept class. Presumably, the more complex a class is, the more difficult it is to learn this class.

The importance of the VC dimension for PAC Learning was discovered by Vapnik and Chevonenkis and by Blumer et al., who proved that a concept class is PAC-learnable if and only if its VC dimension is finite, and that

$$m = O\left(\frac{VC(C)}{\epsilon} \log \frac{1}{\delta\epsilon}\right)$$

samples suffice. An immediate corollary is that every finite concept class is PAC-learnable.

3.2 Double sampling

In this section we describe the "double sampling" argument, which yields the sample complexity as a function of the VC dimension.

We use the following simple lemma.

Lemma 13. *Let $(\Omega, \mathcal{F}, \mu)$ and $(\Omega', \mathcal{F}', \mu')$ be countable probability spaces. Let*

$$F_1, F_2, F_3 ... \in \mathcal{F}, \ F'_1, F'_2, F'_3 ... \in \mathcal{F}'$$

be so that $\mu'(F'_i) \geq 1/2$ for all i. Then

$$\mu \times \mu' \left(\cup_i F_i \times F'_i \right) \geq \frac{1}{2} \mu \left(\cup_i F_i \right).$$

Proof. Let $F = \cup_i F_i$. For every $\omega \in F$, *let* $F'(\omega) = \cup_{i:\omega \in F_i} F'_i$. As there exists i s.t. $\omega \in F_i$ it holds that $F'_i \subseteq F'(\omega)$ and hence $\mu'(F'(\omega)) \geq 1/2$. Thus,

$$\mu \times \mu' \left(\cup_i F_i \times F'_i \right) = \sum_{\omega \in F} \mu(\{\omega\}) \cdot \mu'(F'(\omega)) \geq \sum_{\omega \in F} \mu(\omega)/2 = \mu(F)/2.$$

□

Theorem 14. *Let X be a countable[1] set and $C \subseteq 2^X$ be a concept class of VC-dimension d. Let μ be a distribution over X. Let $\epsilon, \delta > 0$ and m an integer satisfying $2(2m+1)^d(1-\epsilon/4)^m < \delta$. Let $c \in C$ and $Y = (x_1, \ldots, x_m)$ be a multiset of m independent samples from μ. Then, the probability that there is $c' \in C$ so that $c|_Y = c'|_Y$ but $\mu(\{x : c(x) \neq c'(x)\}) > \epsilon$ is at most δ.*

Remark 15. *For a finite concept class, an upper bound of order $\frac{1}{\epsilon} \ln(\frac{|C|}{\delta})$ samples follows by Chernoff's bound and the union bound. Roughly speaking, the double sampling argument allows to replace $\ln |C|$ by d.*

Proof. Let $Y' = (x'_1, \ldots, x'_m)$ be another m independent samples from μ, chosen independently of Y. Let

$$H = \{h \in C : dist_\mu(h, c) > \epsilon\},$$

where

$$dist_\mu(h, c) = \mu(\{x : h(x) \neq c(x)\}).$$

For $h \in C$, define the event

$$F_h = \{Y : c|_Y = h|_Y\}$$

[1] Otherwise we need to assume some measurability conditions.

and let $F = \cup_{h \in H} F_h$. Our goal is thus to upper bound $\Pr_{\mu^m}(F)$. For that, we also define the independent event

$$F'_h = \{Y' : dist_{Y'}(h, c) > \epsilon/2\}$$

where

$$dist_{Y'}(h, c) = \frac{1}{m} \left|\{x' \in Y' : h(x') \neq c(x')\}\right|.$$

We first claim that $\Pr(F'_h) \geq 1/2$ for all $h \in H$. This follows from Chebyshev's inequality: For every $i \in [m]$, let V_i be the indicator variables of the event $h(x'_i) \neq c(x'_i)$ (i.e. $V_i = 1$ iff $h(x'_i) \neq c(x'_i)$). The event F'_h is equivalent to $V = \sum_i V_i/m > \epsilon/2$. Since $h \in H$, we have $p := E[V] > \epsilon$. Since elements of Y' are chosen independently, it follows that $\mathrm{Var}(V) = p(1-p)/m$. Thus, the probability of the complement of F'_h satisfies

$$Pr(F'^c_h) \leq Pr(|V - p| \geq p - \epsilon/2) \leq \frac{p(1-p)}{(p-\epsilon/2)^2 m} < \frac{4}{\epsilon m} \leq 1/2.$$

We now give an upper bound on $\Pr(F)$. We note that By Lemma 13,

$$\Pr(F) \leq 2 \Pr\left(\cup_{h \in H} F_h \times F'_h\right).$$

Let $S = Y \cup Y'$, where the union is as multisets. Conditioned on the value of S, the multiset Y is a uniform subset of half of the elements of S. Thus, by the union bound,

$$\begin{aligned}
2 \Pr\left(\cup_{h \in H} F_h \times F'_h\right) &= 2\mathbb{E}_S \left[\mathbb{E}\left[1_{\{\exists h \in H : h|_Y = c|_Y, dist_{Y'}(h,c) > \epsilon/2\}} | S\right]\right] \\
&= 2\mathbb{E}_S \left[\mathbb{E}\left[1_{\{\exists h' \in H|_S : h'|_Y = c|_Y, dist_{Y'}(h',c) > \epsilon/2\}} | S\right]\right] \\
&\leq 2\mathbb{E}_S \left[\sum_{h' \in H|_S} E\left[1_{\{h'|_Y = c|_Y, dist_{Y'}(h',c) > \epsilon/2\}} | S\right]\right].
\end{aligned}$$

Note that if $dist_{Y'}(h', c) > \epsilon/2$ then $dist_S(h', c) > \epsilon/4$, and hence the probability that we choose Y such that $h'|_Y = c|_Y$ is at most $(1 - \epsilon/4)^m$. Using Lemma 7, we get

$$\Pr(F) \leq 2\mathbb{E}_S \left[\sum_{h' \in H|_S} (1 - \epsilon/4)^m\right] \leq 2(2m+1)^d (1 - \epsilon/4)^m$$

□

Chapter 4

Cover numbers

Scribe: Vsevolod Rakita

In this section we go back to studying the metric properties of VC classes.

Definition 16 (Normalised Hamming Metric)**.**

$$dist(c_1, c_2) = \frac{|\{x \in [n] | c_1(x) \neq c_2(x)\}|}{n}.$$

We specifically study the cover number of VC classes, and prove Haussler's theorem which is an essentially tight upper bound on the cover number. Recall that in previous section we used the probabiitemizeic method to prove the following upper bound, which is not tight.

Claim 17. *Let $\epsilon \in (0, 1]$. Assume that C has $VC(C) = d$. Assume that $dist(c_1, c_2) \geq \epsilon$ for all $c_1 \neq c_2$ in C. Then,*

$$|C| \leq \left(\frac{100 \log(2/\epsilon)}{\epsilon}\right)^d.$$

In this section, we prove a stronger bound:

Theorem 18 (Haussler)**.** *Let $\epsilon \in (0, 1]$. Assume that C has $VC(C) = d$. Assume that $dist(c_1, c_2) \geq \epsilon$ for all $c_1 \neq c_2$ in C. Then*[1]*,*

$$|C| \leq \left(\frac{100}{\epsilon}\right)^d.$$

The theorem was first proved by Haussler and we present a simplification of the proof by Chazelle. The theorem has a geometric interpretation:

[1]The 100 can be made smaller.

Definition 19 (Packing number)**.** *We say C' is an ϵ-packing if for any $dist(c_1, c_2) \geq \epsilon$ for every $c_1 \neq c_2$ in C'. The ϵ-packing number of a set C is the cardinality of the largest $C' \subseteq C$ that is an ϵ-packing.*

Definition 20 (Cover number)**.** *We say C' is an ϵ-cover for C if for every $c \in C$ there is $c' \in C'$ so that $dist(c, c') \leq \epsilon$. The ϵ-cover number of a set C, denoted $N(C, \epsilon)$, is the cardinality of the smallest $C' \subseteq C$ that is an ϵ-cover for C.*

A maximal ϵ-packing is also an ϵ-cover, but the converse is not true (why?).

The theorem thus states that if $VC(C) = d$ then $N(C, \epsilon) \leq (100/\epsilon)^d$. This indicates that in a metric sense a class of VC dimension d behaves like d-dimensional euclidean space.

4.1 Preliminaries

Definition 21 (1-Inclusion graph)**.** *The 1-inclusion graph of C is an undirected graph with vertex set C and edges between c, c' iff they differ in exactly one coordinate.*

These graphs have a useful property which we will use later on.

Lemma 22. *If $VC(C) = d$, then $|E| \leq d|C|$.*

Proof. We will use the technique of shifting. Recall that we proved that if we transform a set C to a set C_{end} by repeatedly applying shifting, then $VC(C_{end}) \leq VC(C)$, and C_{end} is a subset of a Hamming ball of radius d. Hence, any vertex in the 1-inclusion graph of C_{end} has at most d neighbours with a smaller number of 1's (why?). Denote the 1-inclusion graph of C_{end} by (C_{end}, E_{end}). This implies that $|E_{end}| \leq d|C_{end}| = d|C|$.

It remains to prove that $|E| \leq |E_{end}|$. Indeed, we shall show that every shifting operation does not reduce the number of edges in the graph: Let $\{u, v\} \in E$ be an edge, and suppose we want to shift the $x \in [m]$ coordinate. If we shifted both u and v or neither, then the edge is still in the graph. Otherwise, assume we shifted u but not v. Denote the shifted u by u'. Thus, $1 = u(x) = v(x)$, and there is a $x \neq y \in [m]$ so that $u(y) \neq v(y)$. Since v is not shifted, $v' = (v(1), \ldots, 1 - v(x), \ldots, v(m))$ is a vertex as well. So, after the shifting there is an edge between u', v' instead of the edge between u, v. □

Lemma 23. *If $G = (V, E)$ is an undirected graph so that for all $V' \subseteq V$ the induced subgraph on V' contains at most $d|V'|$ edges, then we can direct the edges in E so that every vertex $v \in V$ has $deg_{out}(v) \leq d$.*

Proof. For $v \in V$, denote by $v^1, v^2, \ldots, v^d$ copies of v. For all $\{u, v\} = e \in E$, define $V_e = \{v^1, v^2, \ldots, v^d, u^1, \ldots, u^d\}$. Consider the bipartite graph $G' = (A, B, E')$, where $A = \{V_e : e \in E\}$, $B = \bigcup_i \{v^i : v \in V\}$, and $\{V_e, v^i\} \in E'$ iff $v^i \in V_e$.

For $\tilde{E} \subseteq E$, denote by $U = U(\tilde{E})$ the set of vertices that are touched by edges in $\tilde{E}$, and by $I = I(\tilde{E})$ the edges in the subgraph induced on U in G. Thus, $\tilde{E} \subseteq I$, and

$$|\{V_e : e \in \tilde{E}\}| = |\tilde{E}| \leq |I| \leq d|U| \leq \left|\bigcup_{e \in \tilde{E}} V_e\right|.$$

Hence, by Hall's theorem, there exists a matching from A to B in G'. For every edge $\{u, v\} \in E$, direct it as (u, v) iff $V_{\{u,v\}}$'s match is in $\{u^1, \ldots, u^d\}$. □

4.2 Proof

Finally, we can prove the theorem.

Proof. We actually prove that the size of every ϵ-packing V in C is small.

Easy case: If $m \leq 4d/\epsilon$, then by Lemma 7,

$$|V| \leq |C| \leq \left(\frac{me}{d}\right)^d \leq \left(\frac{4e}{\epsilon}\right)^d,$$

as needed.

Hard case: Assume now that $4d/\epsilon < m$. Let $n = \lceil 4d/\epsilon \rceil$. For $I \subseteq [m]$ and $V \subseteq C$, denote the 1-inclusion graph of $V|_I$ by G_I. Define the function q_I on the vertices of G_I by

$$q_I(f) = |\{g \in V : g|_I = f\}|.$$

Define the function w_I on the edges of G_I by

$$w_I(\{f, g\}) = \frac{1}{(1/q_I(f)) + (1/q_I(g))} \leq \min\{q_I(f), q_I(g)\}.$$

Finally, define

$$W_I = \sum_{e \in E_I} w_I(e).$$

We prove the theorem based on the following properties, which we will prove separately (in the two lemmas below):

1. Almost surely, $W_I \leq d|V|$.
2. $\mathbb{E}W_I \geq 2d(|V| - (en/d))^d$.

These inequalities complete the proof. □

Lemma 24. *Almost surely,* $W_I \leq d|v|$.

Proof. By Lemma 23, we may direct the edges of G_I so that $deg_{out}(u) \leq d$ for all $u \in V|_I$. Thus,

$$W_I = \sum_{e \in E_I} w(e) \leq \sum_{e \in E_I} \min\{q_I(u), q_I(v)\} \leq \sum_{(u,v) \in E_I} q_I(u) \leq d \sum_{u \in V|_I} q(u) = d|V|.$$

□

Lemma 25. $\mathbb{E}W_I \geq 2d(|V| - (en/d)^d)$.

Proof. Let $I = \{i_1, \ldots, i_n\}$. Partition the edges of G_I to n disjoint subsets $E_1, \ldots, E_n$, where the edges of E_j are those who connect vertices that differ in the i_j'th coordinate. Let

$$W_I(j) = \sum_{e \in E_j} w_I(e).$$

Thus, by symmetry,

$$\mathbb{E}W_I = n\mathbb{E}W_I(n).$$

Let $J = I - \{i_n\}$, and for now condition on the value of J, and let i_n be random in $[m] - J$. Partition V into $|V|_J|$ sets

$$V_t = \{g \in V : t|_J = g|_J\}.$$

Partition each V_t into two random sets

$$A_t = \{f \in V_t : f(i_n) = 1\} \text{ and } B_t = V_t - A_t.$$

Let $a_t = |A_t|$ and $b_t = |B_t|$. For every edge $e \in E_n$,

$$w_I(e) = \frac{a_t b_t}{|V_t|}.$$

Thus,

$$\mathbb{E}[W_n|J] = \sum_{t \in V_{|J}} \frac{1}{|V_t|} \mathbb{E}[a_t b_t | J].$$

The number $a_t b_t$ is the number of pairs $f, g \in V_t$ so that $f(i_n) \neq g(i_n)$. Given $f, g \in V$ so that $f_{|J}, g_{|J} \in V_t$, the probability that f and g differ on i_n is at least $\frac{m\epsilon}{m - n + 1} \geq \epsilon$, since V is an ϵ-packing. So every such pair contributes at least ϵ to $a_t b_t$. Since there are

$\binom{|V_t|}{2}$ such pairs in all,

$$\mathbb{E}[a_t b_t | J] \geq \binom{|V_t|}{2}\epsilon.$$

Hence,

$$\mathbb{E}[W_n | J] \geq \sum_{t \in V_{|J}} \frac{\epsilon(|V_t| - 1)}{2} = \frac{\epsilon(|V| - |V_{|J}|)}{2}$$

By Lemma 7,

$$|V_{|J}| \leq \left(\frac{en}{d}\right)^d.$$

So $\mathbb{E}[W_n | J] \geq \dfrac{\epsilon(|V| - (en/d)^d)}{2}$.

To conclude, taking an average over J,

$$\mathbb{E}W = n\mathbb{E}[\mathbb{E}[W_n | J]] \geq \frac{\epsilon(|V| - (en/d)^d)n}{2} \geq 2d(|V| - (en/d)^d)$$

□

Chapter 5

Majority vote game

Scribe: Ghadeer Abu Hariri

The aim of this part is presenting Freund's majority-vote game, and a strategy that guarantees for one of the players a large reward. This game is underlying procedures for boosting the accuracy of learning algorithms. We also discuss implications of the analysis of the majority-vote game to threshold circuits.

5.1 The majority-vote game

The game is played by two players: the weighter and the chooser. The game is played over a set X and comes with a parameter $\gamma > 0$. The game proceeds in iterations. In each iteration i:

- The weightor picks a weight function (probability distribution) W_i on X.
- The chooser selects a set $U_i \subseteq X$ such that $W_i(U_i) \geq 1/2 + \gamma$ and "marks" the points of this set .

Theses iterations are repeated until the weightor decides to stop. The goal of the weighter is to force the chooser to mark each point in the space in the majority of the iterations. The winning set X^* is the set of $x \in X$ that were marked in more than half of the iteration. The overall gain of the weightor is determined by a value function (probability distribution) V on X. The value of the game is $V(X^*)$.

Boosting. Boosting is a method to combine one or several weak learners to a strong learner. In other words, it is about increasing the accuracy of learning procedures. The above game is a key piece of achieving boosting. Roughly speaking, the chooser

represents the weak learning algorithms that errs on $\frac{1}{2}+\gamma$, and the weightor represents a way to combine the choice of the weak learner to a more accurate result that is correct on $1-\epsilon$ of the space.

The strategy. We presents a strategy that lets the weightor gain $1-\epsilon$ in $\lceil \frac{1}{2\gamma^2} \ln \frac{1}{2\epsilon} \rceil$ iterations. Some intuition for the strategy: If we were at the iteration before the last, then we would put the weight only on the part of X for which the last mark will make a difference.

We use the following notation. The integer k is the total number of iterations the game is played. The set X_r^i is the set of points in X that have been marked r times in the first i iterations. The set M_r^i is the subset of X_r^i that is marked in iteration i. Let $q_r^i = V(X_r)^i$ and $x_r^i = \frac{M_r^i}{V(X_r^i)}$. The loss set $L = X - X^*$. From the intuition described above comes the weighting factor that is defined inductively as follow:

$$\alpha_r^{k-1} = \begin{cases} 1 & r = \lfloor \frac{k}{2} \rfloor, \\ 0 & otherwise, \end{cases}$$

and for $0 \leq i \leq k-2$,

$$\alpha_r^i = ((1/2)-\gamma)\alpha_r^{i+1} + ((1/2)+\gamma)\alpha_{r+1}^{i+1}.$$

The performance of this weighting strategy is describe be the following theorem.

Theorem 26. *Assume the weightor plays the majority-vote game for k iterations with k so that*

$$\sum_{j=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{j} ((1/2)+\gamma)^j ((1/2)-\gamma)^{k-j} \leq \epsilon \tag{5.1}$$

and uses weights W_i defined by

$$W_i(A) = \sum_{r=0}^{i} V(A \cap X_r^i)\alpha_r^i / Z_i, \tag{5.2}$$

where set $A \subseteq X$ and $Z_i = \sum_{r=0}^{i} V(X_r^i)\alpha_r^i$ is the normalization. Then the reward at the end of the game is at least $1-\epsilon$.

In order to understand the theorem better, we mention that a simple calculation implies that (5.1) holds whenever

$$k \geq \frac{1}{2\gamma^2} \ln \frac{1}{2\epsilon}.$$

To prove the theorem, we inductively define a potential β_r^i function of the set X_r^i. The potential measures the expected loss from a given state. Define

$$\beta_r^k = \begin{cases} 0 & , r > \frac{1}{2}, \\ 1 & r \leq \frac{1}{2}, \end{cases}$$

and for $i < k$, define

$$\beta_r^i = \left(\frac{1}{2} - \gamma\right)\beta_r^{i+1} + \left(\frac{1}{2} + \gamma\right)\beta_{r+1}^{i+1}. \tag{5.3}$$

A closed formula for β_r^i is given by the tail of the binomial distribution:

$$\beta_r^i = \sum_{j=0}^{\lfloor \frac{k}{2} \rfloor - r} \binom{k-i}{j} \left(\frac{1}{2} + \gamma\right)^j \left(\frac{1}{2} - \gamma\right)^{k-j-i}.$$

The weight function α_r^i is, in some sense, a derivative of the potential function:

$$\alpha_r^i = \beta_r^{i+1} - \beta_{r+1}^{i+1}. \tag{5.4}$$

Another useful property is that β_0^0 equals the left hand side of (5.1). The most important property of the potential is given by the following lemma.

Lemma 27. *For any strategy of the chooser,*

$$\beta_0^0 \geq \sum_{r=0}^{1} q_r^1 \beta_r^1 \geq \sum_{r=0}^{2} q_r^2 \beta_r^2 \geq \ldots \geq \sum_{r=0}^{k} q_r^k \beta_r^k.$$

The lemma yields $\beta_0^0 \geq \sum_{r=0}^{k} \beta_r^k q_r^k$. In Theorem 26 we have $\beta_0^0 \leq \varepsilon$ and $\sum_{r=0}^{k} \beta_r^k q_r^k$ is one minus the value of the game, and so Theorem 26 follows from the lemma.

Proof of Lemma 27. According to the definitions above, for $1 \leq r \leq i$,

$$q_r^{r+1} = q_{r-1}^i x_{r-1}^i + q_r^i (1 - x_r^i)$$

for $r = 0$,

$$q_0^{i+1} = q_0^i (1 - x_0^i)$$

and for $r = i + 1$,

$$q_{i+1}^{i+1} = q_i^i x_i^i.$$

Thus,

$\sum_{r=0}^{i+1} q_r^{i+1}\beta_r^{i+1} = q_0^i(1-x_0^i)\beta_0^{i+1} + \sum_{r=1}^{i}\left(q_{r-1}^i x_{r-1}^i + q_r^i(1-x_r^i)\right)\beta_r^{i+1} + q_i^i x_i^i \beta_{i+1}^{i+1}$.

Rearranging we get

$$\sum_{r=0}^{i+1} q_r^{i+1}\beta_r^{i+1} = \sum_{r=0}^{i} q_r^i\left(1-x_r^i\right)\beta_r^{i+1} + x_r^i\beta_{r+1}^{i+1}\big) \tag{5.5}$$

$$= \sum_{r=0}^{i} q_r^i\beta_r^{i+1} + \sum_{r=0}^{i} q_r^i x_r^i\left(\beta_{r+1}^{i+1} - \beta_r^{i+1}\right). \tag{5.6}$$

The weight restriction implies

$$\sum_{r=0}^{i} W(M_r^i) \geq \frac{1}{2} + \gamma.$$

By definition of the weight function,

$$\frac{1}{Z_i}\sum_{r=0}^{i} V(M_r^i)\alpha_r^i \geq \frac{1}{2} + \gamma.$$

and

$$\frac{\sum_{r=0}^{i} q_r^i x_r^i \alpha_r^i}{\Sigma_{r=0}^{i} q_r^i \alpha_r^i} \geq \frac{1}{2} + \gamma.$$

By (5.4), and since $\beta_r^{i+1} > \beta_{r+1}^{i+1}$,

$$\sum_{r=0}^{i} q_r^i x_r^i(\beta_{r+1}^{i+1} - \beta_r^{i+1}) \leq \left(\frac{1}{2}+\gamma\right)\sum_{r=0}^{i} q_r^i(\beta_{r+1}^{i+1} - \beta_r^{i+1}).$$

Hence,

$$\begin{aligned}\sum_{r=0}^{i+1} q_r^{i+1}\beta_r^{i+1} &\leq \sum_{r=0}^{i} q_r^i\beta_r^{i+1} + \left(\frac{1}{2}+\gamma\right)\sum_{r=0}^{i} q_r^i(\beta_{r+1}^{i+1} - \beta_r^{i+1}) \\ &= \sum_{r=0}^{i} q_r^i\left(\left(\frac{1}{2}+\gamma\right)\beta_{r+1}^{i+1} + \left(\frac{1}{2}-\gamma\right)\beta_r^{i+1}\right) \\ &= \sum_{r=0}^{i} q_r^i\beta_r^i,\end{aligned}$$

where the last equality uses (5.3). Finally,

$$\sum_{r=0}^{i+1} q_r^{i+1}\beta_r^{i+1} \leq \sum_{r=0}^{i} q_r^i \beta_r^i.$$

□

5.2 The power of majority gates

The analysis of the majority-vote game can be used to prove an interesting result regarding the representation of boolean function as a majority over other boolean functions $f : \{-1,1\}^n \to \{-1,1\}$. Let H be a set of boolean functions. Intuitively, we show that if for any distribution over the domain $\{-1,1\}^n$,there is some function $h \in H$ such that f and h are correlated, then f can be represented as a majority over a small number of functions in H.

Let p be a distribution on $\{-1,1\}^n$. The correlation between f and H with respect to p is

$$cor_p(f,H) = \max_{h\in H} \mathbb{E}_{x\sim p}[f(x)h(x)].$$

The distribution-free correlation is

$$cor(f,H) = \min_p cor_p(f,H).$$

The majority function is defined as follows

$$MAJ(x_1,\ldots,x_k) = sign(\sum_{i=1}^{k} x_i)$$

where

$$sign(x) = \begin{cases} 1 & ,x \geq 0 \\ -1 & ,x < 0. \end{cases}$$

Theorem 28. *Let f be a boolean function over $\{-1,1\}^n$ and H be a set of functions over the same domain. If*

$$k \geq 2\ln(2)\frac{n}{(cor(f,H))^2}$$

then for every x,

$$f(x) = MAJ(h_1(x),\ldots,h_k(x))$$

for some $h_1,\ldots,h_k \in H$.

The proof of is based on the analysis majority-vote game. This application is dis-

covered by Schapire. Goldmann, Hastad and Razborov proved a variant of this theorem using von Neumann's minimax theorem from game theory. Their proof does not show how one can find the functions $h_1, \ldots, h_k$. Shcapire's proof is more constructive.

Chapter 6

Compression schemes for Dudley classes

Scribe: Ori Sberlo

6.1 Compression schemes

We define a certain type of compression called *unlabeled compression* and show that there always exists such compression for the dual class of halfspaces. We extend this result to a wider collection of concept classes called *Dudley classes*, and show they are strongly related to a the notion of sign rank.

Definition 29 (Unlabeled compression scheme)**.** *A size-d unlabeled compression scheme for a class $C \subseteq \mathscr{P}(X) = \{0,1\}^X$ is a mapping*

$$H : \mathscr{P}(X)^{\leq d} \to \{0,1\}^X$$

such that for any finite set of labeled examples $S^{\pm}$, there exist some $A \in \mathscr{P}(X)^{\leq d}$ such that,

$$A \subseteq S \quad \textit{and} \quad S^{\pm} \sqsubseteq H(A).$$

Here $\mathscr{P}(X)^{\leq d}$ are all subsets of X of size at most d, a set of labelled examples $S^{\pm}$ is a pair consisting of a set $S \subseteq X$ and a map from S to $\{+,-\}$, and $f \sqsubseteq g$ means that g extends f.

Remarks:

1. $H(A)$ is not necessarily a concept in C.

2. A is just a set; it does not hold the labels of examples $S^{\pm}$.

3. The compression map is implicitly defined, since we demand that every sample has a reconstruction from a compressed sample.

Example: Let C be the concept class of all intervals in $\mathbb{R}$. Define the reconstruction map by

$$H(g) = \begin{cases} [x,y] & g = \{x,y\} \quad x < y, \\ [x,x] & g = \{x\}, \\ \phi & g = \phi. \end{cases}$$

The compression for sample $S^{\pm}$ is obtained by taking the left most positive sample and rightmost positive sample (take the singleton in case of one positive example, or the empty set if there are no positive samples).

Compactness: The following lemma shows that to prove existence of a compression scheme for a given class it suffices to consider finite subclasses of it [Ben-David and Litman].

Lemma 30. *If every finite subclass of C admits an unlabeled size-d compression scheme then so does C.*

The proof uses the compactness theorem for predicate logic of first-order.

6.2 Halfspaces

Every line partition the plane $\mathbb{R}^2$ to two parts. A halfspace is one of these parts. More generally, every hyperplane partitions $\mathbb{R}^n$ to two parts, and a halfspace is one of these parts. A halfspace h may be specified by a linear inequality, derived from the linear equation that specifies the defining hyperplane:

$$h = \{x \in \mathbb{R}^n : \sum a_i x_i + a_{n+1} \geq 0\}.$$

It is uniquely determined by $a = (a_1, \ldots, a_{n+1}) \in \mathbb{R}^{n+1}$. This a corresponds to the hyperplane that is the boundary of h.

Definition 31 (Dual of halfspaces)**.** *Let $(HS^n)^D$ denote the concept class $\{f_r : \mathbb{R}^{n+1} \to \{0,1\} : r \in \mathbb{R}^n\}$ where f_r is define by*

$$f_r(a) = \begin{cases} 1 & \sum_{i=1}^n r_i a_i + a_{n+1} \geq 0, \\ 0 & \sum_{i=1}^n r_i a_i + a_{n+1} < 0. \end{cases}$$

6.3 A compression scheme

Theorem 32 (Ben-David and Litman)**.** *$(HS^n)^D$ admits unlabeled compression of size n.*

We start with an intuitive description of the compression. Using compactness, we can focus on a finite subclass C of $(HS^n)^D$. A sample from C is a finite sequence of halfspaces labeled according to the way some target point classifies them. Such a sample induces a 'cell' in $\mathbb{R}^n$, where every point in the cell is consistent with all samples. If we manage to "remember" a point in the cell then we can reconstruct our samples. The idea is to fix a point $t \in \mathbb{R}^n$, and find the closest point from t to the cell's boundary. This intuitively defines a map $m_t : \{cells\} \to \{points\}$ which depends on t. We will show that given a generic t, the point acquired by m_t can be described at most n many halfspaces from the given sample. These halfspaces define the compression.

We use the following terminology:

- j-dimensional co-set of $\mathbb{R}^n$ is of the form $y + W$ where $y \in \mathbb{R}^n$ and W is a linear j-dimensional subspace of $\mathbb{R}^n$.
- A hyper-plane is $n-1$ co-set.
- A half space is a set of the form $\{\alpha y : \alpha \in \mathbb{R}, \alpha \geq \beta\} + q$ where $y \in \mathbb{R}^n$ and q is a hyper-plane.
- Denote by $B(s)$ the boundary of s and $B(S) = \{B(s) : s \in S\}$.

Definition 33. *Let t be a point, and let g a closed, convex and non-empty subset of $\mathbb{R}^n$. We denote by $y = m_t(g)$ be the unique point so that $d(t, y) = \min\{d(t, x) : x \in g\}$.*

Definition 34. *Let P be a set of hyper-planes. We say that P is regular if for any $P' \subset P$*

1. *If $|P'| \leq n$ then $\cap P'$ is an $(n - |P'|)$ dimensional plane.*
2. *If $|P'| > n$ then $\cap P' = \emptyset$.*

Definition 35. *Let P be a regular set of hyper-planes. We say $t \in \mathbb{R}^n$ is a separating point if for any P', P'' distinct subsets of P of cardinality at most n,*

$$m_t(\cap P') \neq m_t(\cap P'').$$

Lemma 36. *Let $q' \subset q$ be different co-sets then $T_{q',q} = \{t : m_t(q) \in q'\}$ is a co-set of dimension less than n.*

Sketch. After a shift, $T_{q',q}$ is closed under sums and products by scalars. Points t in $q - q'$ satisfy $m_t(q) = t \notin T$. □

Lemma 37. *If P is a finite regular set of hyper-planes then there exists a point that separates P.*

Proof. We shall show that a generic point is good. For two distinct sets P', P'' of hyperplanes in P, Lemma 36 applied to $q = \cap P'$ and $q' = q \cap (\cap P'')$ implies that the set $S(P', P'') = T_{q',q}$ is a co-set of dimension less than n. Since $\mathbb{R}^n$ can not be covered by a finite union of co-sets of dimension less than n, there is a point t that is outside $\bigcup_{P',P''} S(P', P'')$. Specifically, t separates P. □

Definition 38. *Given a halfspace g the adjacent halfspace is defined to be the complement halfspace with the same boundary. Denote by $g^{(0)}$ the adjacent halfspace of g, and by $g^{(1)}$ the halfspace itself. Given a point y, define g_y the halfspace with the same boundary as g that contains y.*

Definition 39. *For point y and a set of halfspaces G, define $G^{(y)} = \{g \in G : y \in g\}$. Based on y and a separating point t for P, define the partial function Γ_y from the set of hyperplanes to the set of halfspaces:*

$$\Gamma_y(g) = \begin{cases} g_y & y \notin g, \\ (g_x)^{(0)} & y \in g, x = m_t(\cap(G^{(y)} - \{g\})), \\ undefined & otherwise. \end{cases}$$

Definition 40. *A cell is a non-empty subset of $\mathbb{R}^n$ which is the intersection of finitely many halfspaces. A cell of P is a cell of the form $\cap S$ where $B(S) \subset P$ and S has no adjacent hyperplanes.*

Proposition 41. *Let q be a cell of P with $q = \cap S$ and $B(S) \subset P$, and let $y = m_t(q)$ Then,*

1. *$m_t(\cap P^{(y)}) = y$.*
2. *$P^{(y)} \subset B(S)$.*
3. *$s = \Gamma_y(B(s))$ for all $s \in S$.*

Proof.

1. Assume towards a contradiction that $m_t(\cap P^{(y)}) = z \neq y$. So, $d(t, z) < d(t, y)$ since $y \in \cap P^{(y)}$. Since $\cap P^{(y)}$ is convex, it contains the interval $[y, z]$. On the one hand, $[y, z] \cap q = [y, y]$, since along $[y, z]$ getting closer to z means getting closer to t. On the other hand, for any $s \in S$ if $y \in B(s)$ then $[y, z] \cap s = [y, z]$, and if $y \notin B(s)$ then $[y, z] \cap s = [y, y'] \neq [y, y]$. Therefore $[y, z] \cap q \neq [y, y]$, a contradiction.

2. Similarly to as above we may conclude that $m_t(\cap(P^{(y)} \cap B(S))) = y$. Since t separates P and by the above,it holds that $P^{(y)} = P^{(y)} \cap B(S)$. This finishes the second claim.

3. Let $s \in S$. If $y \in B(s)$ then $\Gamma_y(B(s)) = (B(s))_y$. This and $y \in s$ implies $\Gamma_y(B(s)) = s$. Now assume that $y \in B(s)$. By the above, $\Gamma_y(B(s)) = (h(B(s), x))^{(0)}$ with $x = m_t(\cap(P^{(y)} - \{B(s)\}))$. Consider the line interval $[y, x]$. Since $d(t, x) < d(t, y)$, we know $[y, x] \cap q = [y, y]$. On the other hand, for any $s' \in S - \{s\}$, we have $[y, x] \cap s' \neq [y, y]$. Hence $[y, x] \cap s = [y, y]$. Therefore $x \notin s$ which implies $\Gamma_y(B(s)) = ((B(s))_x)^{(0)} = s$.

□

Definition 42. *The concept class C is said to be regular if the following holds. Let S be the set of cells defined by hyperplanes in C.*

- $\mathbb{R}^n, \emptyset \notin S$.
- *S has no adjacent halfspaces.*
- *$B(S)$ is regular.*

Lemma 43. *For every finite set of lines L, finite set of point P, and a halfspace s, there exists a halfspace s' satisfying:*

1. *For every $\ell \in L$, the intersection $B(s') \cap \ell$ is singleton.*
2. *$s' \cap P = s \cap P$.*
3. *$P \cap B(s') = \emptyset$.*

Sketch. s' is obtained from s by a small rotation and shift. □

Lemma 44. *Let C be a finite subclass of $(HS^n)^D$ then there is a regular subclass of $(HS^n)^D$ that contains it.*

Proof. Apply lemma 43 and induction on $|S|$. □

Proof of Theorem 32. By Lemma 30, we may consider only finite subclasses of $(HS^n)^D$. Let C be a finite subclass of $(HS^n)^D$. By Lemma 44 we may assume that C is regular. Set $P = B(S)$ and let t and Γ as we defined. Let S' be samples from C and $f : S' \to \{0, 1\}$ denote its labeling. The set $S'' = \{s^{f(s)} : s \in S'\} \subset S'$ induces the cell $q = \cap S''$ of P. The cell is not empty since there must be a point consistent with the samples.

Let $y = m_t(q)$. By proposition 41 Γ_y reconstructs the sample f. That is, Γ_y outputs a point consistent with f, and y is determined from $S^* = (B(S))^{(y)}$ which is of size at most n due to regularity.

6.4 Dudley classes

Here we define a more abstract collection of concept classes for which the compression we discussed above works. These are concept classes that may be embedded in $(HS^n)^D$.

Definition 45. *Let $\mathscr{F}$ be a collection of real-valued functions over some domain X which is vector space over the reals. Let $h : X \to \mathbb{R}$. A Dudley class is a class $C = \{y_f : f \in \mathscr{F}\}$ with*

$$y_f(x) = \begin{cases} 1 & f(x) + h(x) \geq 0, \\ 0 & f(x) + h(x) < 0. \end{cases}$$

Theorem 46 (Dudley)**.** *The VC dimension of such a class is the linear dimension of the vector space $\mathscr{F}$.*

Definition 47. *Let C, C' be concept classes over domains X, X'. An embedding from C to C' is a pair of functions $\pi : X \to X', \tau : C \to C'$ so that for all $c \in C$ and $x \in X$,*

$$c(x) = 1 \Leftrightarrow (\tau(c))(\pi(x)) = 1.$$

We denote this by $C \preceq_{emb} C'$.

Proposition 48. *Let C, C' be concept classes. If C' has a compression with size k and $C \preceq_{emb} C'$ then C also admits a compression with size k.*

Proof. Define the reconstruction function $H(g)(x) = H'(\pi(g))(\pi(x))$ where H' compression for C' and follow the definition of unlabeled compression. □

Proposition 49. *Let D be a Dudley class with dimension k then $D \preceq_{emb} (HS^n)^D$.*

Proof. Follows from definitions. □

The results above imply that

Theorem 50. *If C is a Dudley class. Then C admits an unlabeled compression of size $VC(C)$.*

6.4.1 Sign rank

In light of Theorem 50 we can ask ourselves what is the minimum dimension k for which concept class C with domain X can be embedded in a Dudley class. That is, what is the minimal k for which there exists a vector space of functions $\mathscr{F} = \{f : X \to \mathbb{R}\}$ of dimension k so that C can be embedded in the Dudley class obtained by $\mathscr{F}$.

Let $C = \{c_i\}_{i=1}^{\ell}$ be a finite concept class over $X = \{x_i\}_{i=1}^{n}$. We are looking for $\mathscr{F} = \{f : X \to \mathbb{R}\}$ and functions $\{g_i\}_{i=1}^{l} \subseteq \mathscr{F}$ so that

$$sign(g_i(x_j)) \begin{cases} 1 & c_i(x_j) = 1, \\ -1 & c_i(x_j) = 0. \end{cases}$$

It is convenient to consider sign matrices instead of boolean one (just replace the 0's by -1's). The equality above is simply $sign(g_i(x_j)) = c_i(x_j)$. Suppose $\{f_i\}_{i=1}^{d}$ is basis for $\mathscr{F}$ so that $g_i = \sum \alpha_{i,j} f_j$. Consider the matrices $A_{i,j} = f_j(x_i)$ and $B_{i,j} = \alpha_{i,j}$. Thus $M_{i,j} = (AB)_{i,j} = g_i(x_j)$. Conversely, given a factorization $M_{i,j} = (AB)_{i,j}$ so that $sign(M_{i,j}) = C$, we can obtain $\mathscr{F}$ and $\{g_i\}_{i=1}^{l}$ as required.

Definition 51. *The sign rank of matrix S is*

$$sign - rank(S) = \min\{rank(M) : sign(M_{i,j}) = S_{i,j}\}.$$

The above reasoning shows that the sign-rank captures the minimal d so that concept class C can be realized as Dudley class with dimension k. The following shows that most finite classes have high sign rank, and so their Dudley dimension is large as well.

Proposition 52. *The number of $N \times N$ sign matrices of sign rank at most r does not exceed $2^{O(rN \log N)}$.*

The proof uses real algebraic geometry. Let $P = (P_1, \ldots, P_m)$ be a vector of real polynomials in ℓ variables. Define the semi-variety

$$V(P) = \{x \in \mathbb{R}^{\ell} : \forall i \in [\ell] \;\; P_i(x) \neq 0\}.$$

The sign-pattern of P at x

$$S_P(x) = (sign(P_1(x)), \ldots, sign(P_n(x))).$$

Let $N(P)$ be the number of all possible sign-patterns for P. Notice that $N(P)$ is bounded by the number of connected components of $V(P)$.

Theorem 53 (Warren)**.** *Let $P_1, \ldots, P_m$ be m real polynomials, each in ℓ variables and degree at most k. If $m \geq \ell$ then the number of connected components of $V(P)$ is at most $(4ekm/\ell)^{\ell}$.*

Proposition 52 follows from that the number of sign-patterns possible for real $N \times N$ with rank at most r is bounded by $2^{O(rN \log N)}$. Indeed, a matrix $M \in \mathbb{R}^{N \times N}$ has degree at most r iff it can be factorized into two matrices A, B of sizes $N \times r, r \times N$. Hence the

entries of M can be thought as polynomials in $2Nr$ variables and degree 2. Theorem 53 implies $N(P) \leq (8eN^2/(2Nr))^{2Nr} = 2^{O(rN \log N)}$.

Corollary 54. *For sufficiently large N there exist $N \times N$ matrices with sign rank at least $N/\log N$.*

Chapter 7

Sample compression schemes

Scribe: Vered Cohen

7.1 Definition

Let $C \subseteq \{0,1\}^X$ be a concept class. For $c \in C$ and $Y \subseteq X$, denote by $Y^{\pm,c}$ the element of Y together with the labeling according to c.

A sample compression scheme of size at most d for a concept class $C \subseteq \{0,1\}^X$ consists of a compression function κ and a reconstruction function ρ. The compression function κ maps every finite sample set $Y^{\pm,c}$ to a compression set; a subset of at most d of the labeled samples. The reconstruction function ρ maps every possible compression set to hypothesis $h \in \{0,1\}^X$ such that for all finite $Y \subseteq X$ and $c \in C$, if $h = \rho\left(\kappa\left(Y^{\pm,c}\right)\right)$ then $h(y) = c(y)$ for all $y \in Y$. The hypothesis h is not required to be in C.

Examples:

1. Axis parallel rectangles in $\mathbb{R}^2$. For every sample set of finite size m, save at most 4 points: leftmost, rightmost top and bottom positive samples from our set. The reconstruction function outputs as an hypothesis the smallest rectangle that includes all the points in the compression set.

2. n intervals on a line. Save at most $2n$ points $x_1, x_2, \ldots$: the first positive sample, the first negative example after that, positive after that and so on. The reconstruction function maps the compressed set to the hypothesis that consists of the intervals $[x_1, x_2), [x_3, x_4), \ldots$

It is possible to extend the definition of a sample compression scheme by allowing the compression map to save some extra amount of information. Let Q be a finite set. An extended compression scheme of size d with side information Q for C consists of a compression function κ and a reconstruction function ρ such that the following holds. The compression map κ maps $Y^{\pm,c}$ to a pair (Y', q) where Y' is a subsample of $Y^{\pm,c}$ of size at most d and $q \in Q$. The reconstruction map ρ maps pairs of the form (Y', q) to h. The same correctness should hold.

Example: A shape in $\mathbb{R}^2$ that is either a triangle or an axis-parallel rectangle. The compression function tries to find the "correct" shape: First, try to match with a triangle and then with a rectangle. The side information $q \in \{1, 2\}$ encodes which of the 2 options occurred.

7.2 Learning using a sample compression scheme

Online versus batch learning. An online learning algorithm updates its hypothesis after every sample it receives, whereas a batch-learning algorithm studies a whole 'batch' of examples, after which it produces its hypothesis.

An example for an online learning algorithm. The Halving algorithm keeps track of all concepts consistent with all past examples, and for each new example it predicts the value as the majority of the concepts consistent so far. It then updates the consistent concepts set. This algorithm makes at most $\lg |C|$ mistakes. This construction also yields a a sample compression scheme of size at most $\ln |C|$ for every finite class C.

Batch learning and compression schemes. The batch-learning algorithm corresponding to a sample compression scheme is the straightforward application of it; given a set of examples $Y^{\pm,c}$, it produces the hypothesis $h = \rho(\kappa(Y^{\pm,c}))$.

7.2.1 Connection to PAC learning

The following theorem shows that sample compression schemes give PAC learning algorithms.

Theorem 55 (Littlestone and Warmuth, 1986)**.** *Let μ be any probability distribution on a domain X. Let $C \subseteq \{0,1\}^X$ and$c \in C$. Let ρ, κ be a compression scheme for C of size d. Let $Y = (x_1, x_2.., x_m)$ be a set of examples drawn independently according to μ.*

Define $h = \rho\left(\kappa\left(Y^{\pm,c}\right)\right)$. *Then, for every* $m \geq d$ *and* $\epsilon > 0$,

$$P_{\mu^m}\left(\mu\left(\{x : h\left(x\right) \neq c\left(x\right)\}\right) > \epsilon\right) \leq \sum_{i=0}^{d} \binom{m}{i} \left(1-\epsilon\right)^{m-i}.$$

A similar bound holds even when there is side information Q. The probability of error in this case is at most

$$P_{\mu^m}\left(\mu\left(\{x : h\left(x\right) \neq c\left(x\right)\}\right) > \epsilon\right) \leq |Q| \sum_{i=0}^{d} \binom{m}{i} \left(1-\epsilon\right)^{m-i}.$$

Proof. Let T be a subset of $[m]$. let B_T denote all sample sets $Y = \{x_1, \ldots, x_m\}$ such that the hypothesis $h_T = \rho\left(Y^{\pm,c}\right)$ is consistent with the given examples, that is, for every $x \in Y$,

$$h_T\left(x\right) = c\left(x\right).$$

Let U_T denote all samples sets Y such that

$$\mu\left(\{x : h_T\left(x\right) \neq c\left(x\right)\}\right) > \epsilon.$$

Now, the probability to draw a sample that is in $B_T \cap U_T$is at most $\left(1-\epsilon\right)^{m-|T|}$. Indeed, if Y is in $B_T \cap U_T$, then the examples $\left(x_i : i \in [m] - T\right)$ are both consistent with h_T that is ϵ-far from c and consistent with c.

Finally, the probability that the random set Y has a compression set of size at most d and the error is more than ϵ is at most $\sum_{T:|T|\leq d} \mu^m\left(B_T \cap U_T\right)$, which complete the proof. □

The following lemma helps to understand for which values of m the theorem above is meaningful.

Lemma 56. *Let* $0 \leq \epsilon, \delta \leq 1$ *and let* $0 < \beta < 1$. *If*

$$m \geq \frac{1}{1-\beta}\left(\frac{1}{\epsilon}\ln\frac{1}{\delta} + d + \frac{d}{\epsilon}\ln\frac{1}{\beta\epsilon}\right)$$

then

$$\sum_{i=0}^{d} \binom{m}{i} \left(1-\epsilon\right)^{m-i} \leq \delta.$$

Sketch. Can be deduced from that for all $\alpha > 0$, we have $-\ln\alpha - 1 + \alpha m \geq \ln m$, and from that $\sum_{i=0}^{d} \binom{m}{i} \leq (me/d)^d$. □

7.3 Compression schemes for maximum classes

In this section, we describe an optimal construction of a sample compression scheme for maximum classes.

Definition 57 (Maximum concept class)**.** *A concept class $C \subset \{0,1\}^X$ of VC dimension d is called maximum if it is extremal for Sauer's lemma; that is, for every finite subset $Y \subset X$, it holds that*

$$|C|_Y| = \Phi_d(|Y|),$$

where

$$\Phi_d(m) = \begin{cases} \sum_{i=0}^{d} \binom{m}{i} & m \geq d, \\ 2^m & m < d. \end{cases}$$

Example: $C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ is maximum with $VC(C) = 1$.

The construction of sample compression scheme for maximum classes is based on several strong structural results for them that we now describe.

We use the following notation. The x-restriction of C is the concept class

$$C - x = C|_{X-\{x\}}.$$

The x-reduction of C is the concept class

$$C^{\{x\}} = \{c \in C - x : c \cup (x,0), c \cup (x,1) \in C\}.$$

Theorem 58 (Welzl 1987)**.** *A concept class $C \subset \{0,1\}^X$ over a finite domain X such that $VC(C) = d$ is maximum iff $|C| = \Phi_d(|X|)$.*

Sketch. One direction is immediate from definition, and the other direction follows by induction using binomial identities. □

Corollary 59 (Welzl 1987)**.** *For a maximum concept class $C \subset \{0,1\}^X$ over a finite domain X, and for all $x \in X$:*

1. *$C^{\{x\}}$ is a maximum class of VC dimension $d-1$.*
2. *If $|X - \{x\}| \geq d$, then $C - x$ is a maximum class of VC dimension d.*

For $A = \{x_1, .., x_k\} \subset X$, define the A-reduction of C is as

$$C^A = (((C^{\{x_1\}})^{\{x_2\}})...)^{\{x_k\}}.$$

C^A is well-defined, because $(C^{\{x_1\}})^{\{x_2\}} = (C^{\{x_2\}})^{\{x_1\}}$. A labeling $A^{\pm}$ of A is a map from A to $\pm$ For a labeled set $A^{\pm}$ and $c \in C|_{X-A}$, let the extension of c with $A^{\pm}$ be the concept $c_{A^{\pm}} \in \{0,1\}^X$ that is obtained by extending c by $A^{\pm}$. We can also observe that C^A consists of all concepts c in $C|_{X-A}$ such that for any labeling of A, the extension of c to X is in C.

Corollary 60 (Welzl 1987)**.** *If $C \subset 2^X$ is a maximum class with $VC(C) = d$ over a finite domain X, and $A \subset X$ of size $|A| = d$, then there is only one concept in C^A.*

For A of size d, we say that $A^{\pm}$ is a compression set for the concept $c \in C$ if c is the extension of the unique concept in C^A with $A^{\pm}$.

Lemma 61. *Let C be a maximum class such that $VC(C) = d$ over a finite domain X, and let $A \subseteq Y \subseteq X$ such that $|A| = d$. Let $A^{\pm}$ be a compression set of $c \in C|_Y$. Let c' be the extension of the unique concept in C^A using $A^{\pm}$. Then, $c = c'|_Y$.*

Proof. Holds since $(C|_Y)^A = (C^A)|_{Y-A}$. The left set contains $c|_{Y-A}$ and the right set $c'|_{Y-A}$. □

The lemma above together with the following theorem provide a compression scheme for maximum classes. The compression map maps c to a compression set of c, and the reconstruction map extends the compression.

Theorem 62. *Let C be a maximum class such that $VC(C) = d$, over a finite domain X of size $|X| = m$. Assume $m \geq d$. Then, for every $c \in C$, there is a compression set $A^{\pm}$ of size d.*

Before proving the theorem, we describe the compression scheme. The compression function maps $Y^{\pm,c}$ to its compression set in $C|_Y$. The reconstruction function maps $A^{\pm}$ of size d to the unique concept in C^A with $A^{\pm}$ as extension.

Proof. The proof is by induction.

Base case: If $m = d$ then $A^{\pm} = c$. If $d = 0$ then use the empty set as a compression.

Induction step: Let $c \in C - x_m$ Consider two different cases.

Case 1: $c \notin C^{\{x_m\}}$, so only one of $c \cup (x_m, 1)$ and $c \cup (x_m, 0)$ is in C. We know that $C - x_m$ is a maximum class of dimension d, since $m > d$. By induction, c is represented by $A^{\pm}$ of size d with $A \subset X - \{x_m\}$. The extension of the unique concept in $(C|_{X-\{x_m\}})^A$ by A is c and c can be extended in a unique way to all of X.

Case 2: $c \in C^{\{x_m\}}$. We know that $C^{\{x_m\}}$ is a maximum class of dimension $d-1$. By the induction hypothesis, there is a compression set $B^{\pm}$ of size $d-1$ so that $B \subseteq X - \{x_m\}$. Define

$$A^{\pm} = B^{\pm} \cup (x_m, c(x_m)) .$$

We know that $A^{\pm}$ represents a single concept $c_{A^{\pm}}$ in C. We only need to show that $c_{A^{\pm}}|_{X-\{x_m\}} = c_{B^{\pm}}$. Let assume, towards a contradiction, that it is not so. Then, there exists $x \in X - (\{x_m\} \cup B)$ such that $c_{A^{\pm}}(x) \neq c_{B^{\pm}}(x)$. There are 2^d concepts in C where $c(x) = c_{A^{\pm}}(x)$. There are 2^{d-1}concepts in $C^{\{x_m\}}$ where $c(x) = c_{B^{\pm}}(x)$. By the definition of $C^{\{x_m\}}$, there are 2^d concepts in C where $c(x) = c_{B^{\pm}}(x)$. So we have a shattered set of size $d+1$ in C, a contradiction. □

Chapter 8

Population recovery

Scribe: Igor Khmelnitsky

In this chapter we consider two learning problems: the population recovery problem (PRP) and the distribution recovery problem (DRP). We present a solution using PID (partial identification) graphs. These are reconstruction problem investigated by Wigderson and Yehudayoff. Our to reconstruct a set of items or a distribution on them using distorted samples.

Let us start by giving 2 examples of the PRP for lossy and noisy sample[1]:

Example 63 (Recovery from lossy samples)**.** *Imagine that you are a paleontologist, who wishes to determine the population of dinosaurs that roamed the Earth before the hypothetical meteor made them extinct. Typical observations of dinosaurs consist of finding a few teeth of one here, a tailbone of another there, perhaps with some more luck a skull and several vertebrae of a third, and rarely a near complete skeleton of a fourth. Each observation belongs to one of several species. Using these fragments, you are supposed to figure out the population of dinosaurs, namely, a complete description of (say) the bone skeleton of each species, and the fraction that each species occupied in the entire dinosaur population. Even assuming that our probability of finding the remains of a particular species, e.g. Brontosaurus, is the same as its fraction in the population, the problem is clear: while complete features identify the species, fragments may be common to several. Even with knowledge of the complete description of each type (which we a-prior do not have), it is not clear how to determine the distribution from such partial descriptions. A modern-day version of this problem is analyzing the partial Net ix matrix, where species are user types, features are movies, and each user ranked only relatively few movies.*

[1]The text in examples is copied verbatim from paper of Wigderson-Yehudayoff.

Example 64 (Recovery from noisy samples)**.** *Imagine you are a social scientist who wants to discover behavioral traits of people and their correlations. You devise a questionnaire where each yes/no question corresponds to a trait. You then obtain a random sample of subjects to fill the questionnaire. Many of the traits, however, are private, so most people are unwilling to answer such questions truthfully, fearing embarrassment or social consequences if these are discovered.To put your subjects at ease you propose that instead of filling the questionnaire truthfully, they fill it randomly: as follows: to each question, they flip the true answer with probability 0.49, independently of all others. This surveying method (typically applied when there is one question of interest) is known as "randomized response" and was initiated by Warner. Is privacy achieved? Can one recover the original truthful information about the original sequences of traits and their distribution in the population? Observe that typical samples look very different than the sequences generating them. Indeed, can one synthesize sensitive databases for privacy by publishing very noisy versions of their records?*

Both of this examples illustrate the issues we want to tackle, and hopefully its importance.

Set-up. Fix an alphabet Σ so that $? \notin \Sigma$, parameters $n, k \in \mathbb{N}$, accuracy parameter $\alpha > 0$, error margin $\delta > 0$, a vector population $V \subseteq \Sigma^n$ which is of size $|V| \leq k$ and a probability distribution π on V.

Definition 65 (Lossy sample)**.** *Let $0 < \mu < 1$ then a μ-lossy sample v' is generated randomly in the following way. First pick $v \in V$ randomly using the distribution π. Then, independently at random for each coordinate v_i of v replace it with "?" with probability $1-\mu$ and left untouched with the probability μ. The result is v'. For example:*

$$v = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \underset{loss}{\rightarrow} v' = \begin{pmatrix} 1 \\ ? \\ 1 \\ ? \end{pmatrix}$$

Definition 66 (Noisy sample)**.** *Let $-1 < \nu < 1$ and for simplicity assume $\Sigma = \{0, 1\}$. A ν'-noisy sample is generated as follows. First pick $v \in V$ randomly using the distribution π. Then, independently at random each coordinate v_i of v is filliped with the probability $(1-\nu)/2$, and left untouched with the probability $(1+\nu)/2$. The result is v'. For example:*

$$v = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \underset{noise}{\rightarrow} v' = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Now let us formally define the problem we are trying to solve.

Definition 67 (Population recovery problem (PRP))**.** *Approximately reconstruct V and π up to the accuracy parameter α with probability δ for error. Namely, given an independent lossy or noisy samples, find a set V' which contains every element $v \in V$ for which $\pi(v) > \alpha$ and a distribution π' on V' so that that for each $v \in V'$ we have $|\pi(v) - \pi(v')| < \alpha$. We want to find such V', π' with the probability at least $1 - \delta$.*

In this presentation we focus on a simpler problem the distribution recovery problem.

Definition 68 (Distribution recovery problem (DRP))**.** *Same as PRP, but we know the vector set V, so the only thing left to reconstruct is π.*

The solution of DRP will already contains all ideas needed to solve PRP. To solve DRP, we shall introduce the notion of Partial IDs (PIDs) and describe a useful construction of PIDs. Later on, we shall hint how the DRP problem is solved using PIDs.

8.1 Partial IDs

When trying to solve the PRP problem we notice that we do not always need all the information about a certain item to recognize it, most of the time a partial information is enough. For example in a set of family members we do not always need all the info about a certain member (ID number, name, age, ...) to identify her, and usually just the name is enough. Unfortunately, sometime different people may look the same, when only a partial information is given. In our family example a father and his child can both be named George. Lets us define a PID in our case:

Definition 69 (Partial ID (PID))**.** *Let $V \subseteq \Sigma^n$. For $v \in V$ let $S \subseteq [n]$. We think of S as a partial ID for v. The restriction of v to S is denoted by $v[S]$. An impostor $u \in V$ of v is a vector so that $u[S] = v[S]$. Denote by $I(v; S)$ the set of all imposters of v with respect to S.*

A natural way to record the imposter relations is in a form of a directed graph. We shall think of the items with their PID as vertices, where each v points on all of his imposters.

Definition 70 (PID graph)**.** *A PID graph G is defined for any set of vectors $V \subseteq \Sigma^n$ and any set of PIDs $\mathcal{S} = \{S_v \subseteq [n] : v \in V\}$. The vertices of G are the elements of V. Directed edges go from every vertex v to every importer u of v, namely $v \rightsquigarrow u$ if $u \neq v$ and $u[S_v] = v[S_v]$.*

The following are few properties of the PID graph which will be of use for us later on:

Definition 71. *Basic properties for PID graph G:*

a. *For every $v \in V$, the cost of v is defined inductively as follows. If v is a sink in G then $cost(v) = 1$. For all other v:*

$$cost(v) = 1 + \sum_{u \in I(v): u \neq v} cost(u).$$

The cost of the entire graph is defined to be

$$cost(G) = \max \{cost(v) : v \in V\}.$$

b. *The depth of G is defined to be the longest directed path in G.*

c. *The width of G is defined to be*

$$width(G) = \max \{|S_v| : v \in V\}.$$

We would like to minimize all of three parameter, as they will all effect the efficiency of our algorithms. We shall see that we can create our PID graph in such a way that we keep $width(G) \leq log(|V|)$,$depth(G) < log(|V|)$, and by using the following claim we get that $cost(G) \leq |V|^{log(|V|))}$.

Claim 72. *If a PID graph G is acyclic then $cost(G) \leq |V|^{depth(G)}$.*

Now we explain how to construct an efficient PID graph. We start by describing the following recursive algorithm which we will use to extend a given PID S for a vector v. This algorithm greedily shrinks the number of imposters by a factor of two until it cannot do it anymore. (In the algorithm, the set of vectors V is fixed.)

Algorithm: $Extend(v, S)$

Input: A vector $v \in V$ and a set $S \subseteq [n]$.

Recursion base: Let J be the set of i in $[n] \backslash S$ so that

$$|I(v, S \cup \{i\})| \leq |I(v, S)|/2.$$

If J is empty, output.

$$Extend(v, S) = S.$$

Recursive step: Otherwise let $i = \min J$, and compute

$$Extend(v, S) = Extend(v, S \cup \{i\}).$$

The following claim summarizes the important properties of the Extend algorithm.

Claim 73 (Properties of extension)**.** *For every $v \in V$ and $S \subseteq [n]$, if $T = Extend(v, S)$ then:*

- $S \subseteq T$.
- $|I(v, T)| \leq |I(v, S)| \cdot 2^{|S|-|T|}$.
- *T is maximal for v that is $Extend(v, T) = T$.*
- *If $u \neq v$ and $u \in I(v, T)$, then T is not maximal for u.*

Proof. Verification is an exercise. □

One may think that choosing a PID for v by $S_v = Extend(v, \emptyset)$ would be enough to get our desired object, but unfortunately this is not the case. In particular, although we do make sure that $width(G) \leq \log(|V|)$, we cannot guarantee that the depth of the graph is logarithmic, see example 77 below.

What we can do is ensure that for all $u \neq v$ so that $u \in I(v, S_v)$, we have $|S_u| > |S_v|$, and by doing so get a graph of logarithmic depth as well:

Algorithm: PID construction

Input: A set $V \subseteq \Sigma^n$

Initialize: For every $v \in V$ set $S_v = Extend(v, \emptyset)$.

Iterate: While there exist v and $u \neq v$ with $u \in I(v, S_v)$ and $|S_u| \leq |S_v|$ set $S_u = Extend(u, S_v)$.

Output: The set of final PIDs S_v and the graph G they define.

Before proving that this algorithm does what we want it to do, let us give an example to see it in action:

Example 74. *Set our* $\Sigma = \{0,1\}^5$ *and*

$$V = \begin{array}{r|ccccc} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} \\ v_1: & 1 & 1 & 1 & 1 & 0 \\ v_2: & 0 & 0 & 1 & 0 & 1 \\ v_3: & 1 & 1 & 1 & 1 & 1 \\ v_4: & 1 & 0 & 0 & 1 & 0 \\ v_5: & 1 & 0 & 1 & 0 & 0 \\ v_6: & 1 & 1 & 1 & 0 & 0 \end{array}$$

In the "initialize" phase we get the sub set of coordinates for each $v \in V$*:*

$$S_{v_1} = \{2\}; S_{v_2} = \{1\}; S_{v_3} = \{2,5\}; S_{v_4} = \{2,3\}; S_{v_5} = \{2\}; S_{v_6} = \{2,4\}.$$

With the imposters:

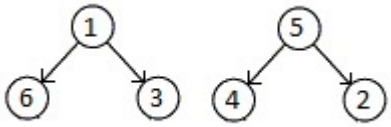
$$I(v_1, S_{v_1}) = \{v_1, v_3, v_6\}; I(v_2, S_{v_2}) = \{v_2\}; I(v_3, S_{v_3}) = \{v_3\}$$

$$I(v_4, S_{v_4}) = \{v_4\}; I(v_5, S_{v_5}) = \{v_2, v_4, v_5\}; I(v_6, S_{v_6}) = \{v_6\}.$$

Now we got to the iteration stage and it is easy to see that only for v_5 *we have* $v_2 \in I(v_5, S_{v_5})$ *and* $|S_{v_2}| \leq |S_{v_5}|$*. Therefore the "iteration" stage is not empty, so we set*

$$S_{v_2} = Extend(v_2, S_{v_5}) = \{2,1\}.$$

And now since we have no other v *which satisfies the condition in the iteration stage we are finished and we get the graph* G*:*



With the properties:

$$Cost(G) = 3; Depth(G) = 2; Width(G) = 2.$$

Now let us prove the following claim which summarizes the properties maintained by the algorithm:

Claim 75 (Properties of the PID graph)**.** .

a. *If $u \neq v$ in V were chosen in some iteration then*

$$|Extend(u, S_v)| > |S_v| \geq |S_u|.$$

b. *The total length of PIDs $\Sigma_{v \in V}|S_v|$ strictly increases at every iteration.*

c. *For every $v \in V$ and every S_v, which is obtained while the algorithm runs, $1 \leq |I(v, S_v)| \leq 2^{-|S_v|} \cdot |V|$. Specifically, the size of S_v never exceeds $\log |V|$.*

d. *The total length of PIDs never exceeds $|V| \log |V|$.*

e. *The algorithm halts after at most $|V| \log |V|$ iterations.*

f. *Let G be the PID graph the algorithm computed. Then, along every path, the size of the corresponding PIDs strictly increases.*

Proof. Verification is an exercise. □

The following theorem summarizes the properties of the construction described above.

Theorem 76. *The algorithm of the PID graph construction terminates in at most $|V| \log |V|$ iterations, and produces a PID graph G with $depth(G) \leq \log k$ and $width(G) \leq \log k$.*

8.1.1 Example for need for several Extends

Here is a set V for which the procedure: for all $v \in V$,

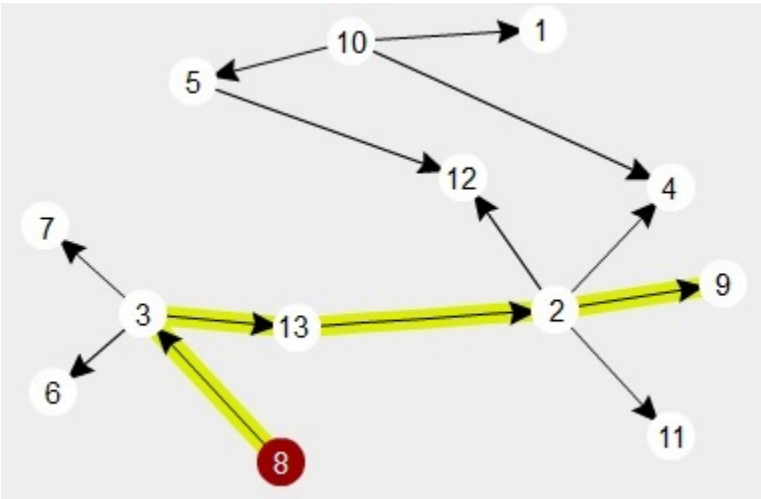
$$S_v = Extend\,(v, \emptyset)$$

does not yield a graph of $depth(G) \leq \log |V|$. We are not going to present all calculations but only show the set V and the created graph.

Example 77. *We have* $n = 5$ *and the following set* V*:*

$$\begin{pmatrix} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} \\ v_1: & 1 & 0 & 0 & 0 & 0 \\ v_2: & 0 & 1 & 1 & 0 & 1 \\ v_3: & 0 & 1 & 0 & 1 & 1 \\ v_4: & 1 & 0 & 1 & 0 & 0 \\ v_5: & 1 & 1 & 0 & 1 & 0 \\ v_6: & 0 & 1 & 0 & 0 & 0 \\ v_7: & 0 & 0 & 0 & 1 & 1 \\ v_8: & 0 & 1 & 0 & 1 & 0 \\ v_9: & 0 & 0 & 1 & 0 & 1 \\ v_{10}: & 1 & 0 & 0 & 1 & 0 \\ v_{11}: & 0 & 1 & 1 & 1 & 0 \\ v_{12}: & 1 & 1 & 1 & 1 & 1 \\ v_{13}: & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

For which the graph we get is:



Where vertex 8 *is points to all the other vertices but only one edge is drawn. One of the longest paths in the graph is highlighted. We can easily see that*

$$depth\,(G) = 4 > \log|V|.$$

8.2 Solving the distribution recovery problem

In this section we outline an algorithm which solves the DRP using the PID graphs we described in the previous section. Recall that we are given a set of vectors V and we

want to reconstruct the probability $\pi(v)$ for every $v \in V$ up to some accuracy parameter α, using the noisy or lossy sampler. The algorithm we shall describe in this chapter is divided into 2 main phases an "estimation" phase and an "aggregation" phase.

The estimation phase is about using enough random sample to estimate the following quantity:

$$p(v) = \sum_{u \in I(v)} \pi(u).$$

Denote the adjacency matrix of the PID graph by M. Thus,

$$p = M\pi.$$

Since M is a triangular matrix with 1s on the diagonal, we can calculate π from p by inverting M. The properties of G imply that the inversion of M is numerically stable, so a small error in our estimation of p will not translate to a huge error in our estimation of π:

$$\|\pi - \pi'\|_\infty \leq cost(G)\|p - p'\|_\infty,$$

where $p = M\pi$ and $p' = M\pi'$. To compute $p'(v)$, our estimation of $p(v)$, we just take enough samples and measure statistics in the small window defined by S_v. Since $|S_v| \leq \log|V|$, this can be done in polynomial time (with high probability).

Chapter 9

Teaching

Scribe: Itai Rosenberg

9.1 Teaching dimension

In many natural learning scenarios, the teacher has an important part of the learning process. However, most algorithmic learning models neglect this aspect. This lecture is about models in which the teacher plays a bigger role.

The teacher or the textbook often has a common objective with the student – that the student will learn quickly as possible the material. As a result, the teacher tries to give helpful information, thereby aiding the student's learning process and definitely not being adversarial.

In terms of the classroom scenario, the task of a teacher is to give the students information that will help all of them learn the subject quickly. Teaching in this sense means giving information about a target object such that all admissible learners identify it.

What is the teaching process? At the beginning of the teaching process, the teacher is given a target object from a class of possible objects. Then, in every round, the teacher selects some piece of information about that object and gives it to a set of learners. Every learner in turn computes an hypothesis based on all information received so far. The process ends when all learners hypothesize the target. The number of rounds until this kind of teaching success is achieved depends on the teacher. The minimum number of rounds taken over all teachers is a measure for the teachability of the target object.

The properties of the learners have a significant effect on the amount of information that the teacher needs to provide that will allow all learners to hypothesize correctly.

The only assumption that we make is that each learner's hypothesis is consistent with all the information that the teacher gave until now.

To make all class hypothesize correctly, the teacher need to give enough information that the target object will be the only object consistent with the information that the teacher gave so far.

Example: $n + 1$ binary strings of length n as follow:

$$\begin{array}{c} 0000...00 \\ 0000...01 \\ 0000...10 \\ \vdots \\ 1000...00 \end{array}$$

If we want to teach the entire class one of the strings containing a 1 it will be very easy. We will just reveal the unique position of the 1. However, teaching the all 0 string is much harder. We will have to reveal all the n bits which takes n rounds. Moreover, the class, although rather simple, has a teaching dimension of n too, which is the highest possible for a class of length n strings.

9.1.1 Notation

A concept class C is a subset of $\{0,1\}^n$. A learning algorithm for C receives a set S of examples for a concept $c \in C$ and computes a hypothesis h. A consistent and class preserving learning algorithm may choose the hypotheses only from the set:

$$H(S) = \{h \in C : \ h \text{ is consistent with } c \text{ on } S\}$$

The set S is called a teaching set for c in C if $H(S) = \{c\}$.

The teaching dimension of c in C is

$$TD(c, C) = \min\{|S| : H(S) = \{c\}\}.$$

The teaching dimension of a concept c specifies the number of examples an optimal teacher needs for teaching c to all learners.

The teaching dimension of C is

$$TD(C) = \max\{TD(c, C) : c \in C\}.$$

Two concepts differing only with respect to one instance $x \in [n]$ are called neighbors. The number of neighbor concepts of c is a lower bound for the teaching dimension of c because each neighbor concept must be ruled out by a separate example.

9.2 Monomials and 2-term DNFs

A monomial (or conjunction of literals) is a function of the form $v_1 \wedge v_3 \wedge \overline{v_5}$. The set of all concepts represented by monomial is denoted by $1 - M_n$. Every monomial, except the contradictory one (the zero function), can be represented by a string $M \in \{0, 1, *\}^n$, where $M[i] = \{0, 1, *\}$ specifies whether the variable v_i occurs negated, unnegated, or not at all. For example, for $n = 3$, the vector 100 represents $v_1 \wedge \overline{v}_2 \wedge \overline{v}_3$. We write $M_1 \subseteq M_2$ when for all i we have either $M_1[i] = M_2[i]$ or $M_2[i] = *$. That is, as function M_2 is at least M_1.

A 2-term DNF is disjunction of at most two monomials $M_1 \vee M_2$. The set of concepts represented by 2-term DNFs is denote by $2 - M_n$. It holds that $1 - M_n \subsetneq 2 - M_n$

Claim 78. *For all non-empty c in $1-M_n$ representable by a monomial with k variables, we have $TD(c, 1-M_n) = \min\{k+2, n+1\}$. The empty concept has a teaching dimension of 2^n because all its 2^n neighbors are contained in $1 - M_n$.*

Sketch. Consider e.g. the string $1^k *^{n-k}$. If $k < n$, then a minimum teaching set contains two complementary positive examples $(1^k 0^{n-k}, 1)$ and $(1^k 1^{n-k}, 1)$ and k negative examples $(1^i 0 1^{k-i-1} 0^{n-k}, 0)$ for $i = 0, \ldots, k-1$. This results in a teaching set with $k+2$ elements. If $k = n$, a minimum teaching set contains the unique positive example $(1^n, 1)$ and the n negative examples $(1^i 0 1^{k-i-1} 0^{n-k}, 0)$ for $i = 0, \ldots, k-1$. This teaching set has cardinality $n + 1$. □

9.2.1 Karnaugh map

The Karnaugh map is a pictorial method to consider boolean algebra expressions. It reduces the need for extensive calculations by taking advantage of our pattern-recognition capability. Boolean expressions are transferred from a truth table to a two-dimensional grid, in which each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. In this representation, minterms must be rectangular and must have an area that is a power of two (i.e., $1, 2, 4, 8, \ldots$). The rectangles chosen should be as large as possible without containing any 0s. Rectangles may overlap in order to make each one larger.

We can use the Karnaugh map to detrmine how many examples (positive and negative) are required to give the learner, so that she will be able to hypothesize the concept.

Example: A teaching set for $11**$ in

	v_1	v_1	$\overline{v_1}$	$\overline{v_1}$	
v_2	1111	1101	0101	0111	v_4
v_2	1110	1100	0100	0110	$\overline{v_4}$
$\overline{v_2}$	1010	1000	0000	0010	$\overline{v_4}$
$\overline{v_2}$	1011	1001	0001	0011	v_4
	v_3	$\overline{v_3}$	$\overline{v_3}$	v_3	

The grey cells are the area that we want that all learners will hypothesize exactly. We need to give 2 positive examples and 2 negative examples that are exactly $k + 2 = 4$:

	v_1	v_1	$\overline{v_1}$	$\overline{v_1}$	
v_2	1		0		v_4
v_2		1			$\overline{v_4}$
$\overline{v_2}$	0				$\overline{v_4}$
$\overline{v_2}$					v_4
	v_3	$\overline{v_3}$	$\overline{v_3}$	v_3	

9.3 Importance of context

The following demonstrates the importance of context. We saw that $1 - M_n$ is easy to teach when the context is $1 - M_n$, but we now see that $1 - M_n$ is hard to teach when the context is $2 - M_n$.

Claim 79. *For $c \in 1 - M_n \setminus \{\{0,1\}^n, \emptyset\}$, we have $TD(c, 2 - M_n) \geq 2^{n-1}$.*

Proof. The concept $c \in 1 - M_n \setminus \{\{0,1\}^n, \emptyset\}$ can be represented as a monomial with at least one variable, and at most n variables. Therefore, the concept c does not contain at least half of the instances, and there are at least 2^{n-1} instances not in c. Let t be a instance so that $t \notin c$. The concept $t \vee c$ is in $2 - M_n$, and $t \vee c$ is a neighbor concept of c. Therefore, c has at least 2^{n-1} neighbor concepts in $2 - M_n$, which proves the claim. □

9.4 Optimal teachers

We can also consider the case where the learners assume that their teacher is optimal. Namely, the teacher does not give superfluous example. This way (as we will see) the number of examples needed for the learners to be successful can be reduced compared to the teaching dimension.

Example: Class S_n over $\{0,1\}^n$ that contains the empty concept c_0 and all the singleton concepts $(c_z : z \in \{0,1\}^n)$. Clearly, $TD(S_n) = TD(c_0) = 2^n$ and $TD(c_z) = 1$.

Now, as we saw before, to teach c_z for $z \in \{0,1\}^n$ is easy. The problem is only when the teacher tries to teach the c_0 concept. If the students do not make any assumption about their teacher, it takes the teacher 2^n rounds to teach c_0. However, if the class assumes the teacher is optimal, and the teacher gave a negative example, the students can know for sure that the teacher is trying to teach them c_0. This gives a much more efficient teaching process (just one example).

The above assumption may be too strong, because it demands that the learner will know for each concept its teaching set. We can demand a more realistic demand that each learner knows only the teaching dimension for each concept. This knowledge allows the learners to ignore all hypotheses whose teaching dimensions are smaller than $|S|$. This allows to teach the previous example using 2 teaching examples instead of 2^n.