CSE 513T Theory of Artificial Intelligence & Machine Learning February 6, 2018

Lecture 7

Lecturer: Alex Durgin *Scribe: Hai Son Le*

Contents

1 Definition of VC Dimension **1**
1.1 Motivation 1
1.2 Definition 1

2 Examples **1**

3 No Free Lunch Theorem **2**

1 Definition of VC Dimension

1.1 Motivation

So far, we have only characterized the learnability of concept classes over the Boolean variables $\{0,1\}^n$. But what about concept classes over numerical inputs (e.g. $X = R^n$)?

Example 1 *n dimensional boxes $[a_1, b_1] \times [a_2, b_2] \times ...[a_n, b_n]$.*
They can be described by only $2n$ parameters. Can we apply our previous strategies to decide its learnability? Is such a class learnable?

The definition of VC Dimension will help us to answer these questions.

1.2 Definition

Definition 2 *For any concept class C over X, and any $S = \{x_1, x_2, ..., x_m\} \subset X$, all the dichotomies realized by C, is defined as:*

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

If $\Pi_C(S) = \{0,1\}^m$(where $m = |S|$), we say that S is shattered by C. Thus, S is shattered by C if C realizes all possible dichotomies of S.

*The **Vapnik-Chervonenkis (VC)** dimension of a representation class C, denoted as VCD(C), is the cardinality d of the largest set S that shattered by C. If any large finite sets can be shattered by C, then VCD(C) $= \infty$. Note that for VCD(C) to be d, we don't need to show that C shatters **every** sample S with size d. Showing it does for one set S is sufficient.*

2 Examples

Example 3 *C is the concept class of all functions: $N \to \{0,1\}$.*
Obviously, $VCD(C) = \infty$ since C can shatter any set of finite sizes.

Example 4 *C is set of intervals on the Real line.*
We see that any possible dichotomies of 2 points can be shattered by an interval. Therefore, the $VCD(C)$ is at least 2. However, a dichotomy of 3 points, for example: $\{+,-,+\}$, can't be shattered by any interval. Therefore, $VCD(C) = 2$.

Example 5 *C is set of halfspaces in R^2.*
We can easily see that any three non-colinear points can be shattered by a halfspace (i.e. a linear separator line). Hence, $VCD(C)$ is at least 3. However, a dichotomy of 3 points, as shown in Figure 1, can't be shattered by any halfspace. Therefore, $VCD(C) = 3$.

In general, VCD of halfspaces in R^n is $n + 1$. (Proof in Homework 4).

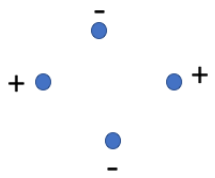

Figure 1: Dichotomy of 4 points that is not shattered by halfspaces

Example 6 *C is set of axis that aligned rectangles in R^2.*
We can easily show that any dichotomy of four points can be shattered by rectangles (by drawing). Hence, $VCD(C)$ is at least 4. However, a dichotomy of 5 as shown in Figure 2, can't be shattered by any rectangle. Therefore, $VCD(C) = 4$.

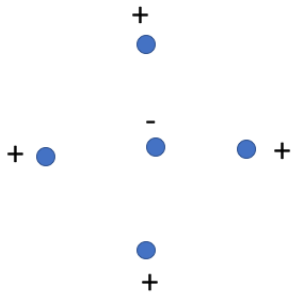

Figure 2: Dichotomy of 5 points that is not shattered by any rectangle

3 No Free Lunch Theorem

Theorem 7 *Let C be a concept class of VC dimension d. Then any algorithm that with probability $> 1/7$ produces h s.t. $Pr_{x \in D}[h(x) = c(x)] \geq 1 - \epsilon$ must use $\Omega(d/\epsilon)$ examples.*

Idea We will find an adversarial distribution for any algorithm that uses too few examples. In particular, we note that there's some set S of size d that c shatters. Hence, there's no bias help given on this set for unknown labels.

Proof
First, we will show that we need $\geq \Omega(d)$ examples and then we'll refine the same argument to tighten this bound up.
Let D be a uniform distribution over d points $\{x_1, ..., x_d\}$. Since $VCD(C) = d$, there exists a set S of d points that can be shattered by C no matter what labels they have. Therefore, all 2^d possible label

combinations are shattered by C.
Consider the following experiment: fix any PAC-learning algorithm for C and let it observe $d/2$ different points from $\{x_1, ..., x_d\}$ to produce a hypothesis h. Then for any x_i not observed in the algorithms set of examples, the hypothesis disagrees with its label with probability 1/2. The expected error of h on D is: $\sum_{i=1}^{d} Pr(x_i)*$(expected error on x_i) $\geq \frac{d}{2} * \frac{1}{d} * \frac{1}{2} = \frac{1}{4}$.
We want to know with what probability our algorithm will output a hypothesis that does better. In particular, we want to know with what probability h can achieve error $\leq \frac{1}{8}$. Let p be that probability. We're going to bound p. We know that: $\frac{1}{4} \leq$ (expected error of h) $\leq \frac{1}{8}p + (1-p) * 1$. Thus, $p < \frac{6}{7}$. Therefore, with probability $p > \frac{1}{7}$, the algorithm will output hypothesis with error $\geq \frac{1}{8}$. This means in order to achieve error $< 1/8$ with probability $> 1/7$, we need to use $\Omega(d/2)$ examples. ■