Introduction to Machine Learning **Fall Semester**

Recitation 3

Lecturer: Regev Schweiger *Scribe: Regev Schweiger*

3.1 VC Dimension

3.1.1 Motivation

We are interested in the question of what hypothesis classes are learnable. One simple result we have seen is this:

Theorem 3.1 *Every finite concept class $\mathcal{H}$, for the case $\mathcal{C} = \mathcal{H}$, is PAC-learnable, with*

$$m(\varepsilon, \delta) \geq \frac{1}{\epsilon} \ln \frac{|\mathcal{H}|}{\delta}.$$

examples sufficient for PAC-learning.

On the other hand, we saw that the class of *all* hypotheses is not PAC-learnable (the No Free Lunch theorem). We would like to be able to handle infinite concept classes, classify which are PAC-learnable, and with what sample complexity.

3.1.2 Definitions

We start with a few definitions. Assume $\mathcal{C}$ is a concept class defined over the instance space $\mathcal{X}$.

Definition For a concept class $\mathcal{C}$ over $\mathcal{X}$ and for any $S = \{x_1, \ldots, x_m\} \subseteq X$, we define the projection of $\mathcal{C}$ on S, $\Pi_{\mathcal{C}}(S) \subseteq \{0,1\}^m$, as:

$$\Pi_{\mathcal{C}}(S) = \{\langle c(x_1), \ldots, c(x_m) \rangle : c \in \mathcal{C}\}$$

We are interested only whether all possible functions on S are realizable:

Definition A concept class $\mathcal{C}$ *shatters* S if $2^{|S|} = |\Pi_{\mathcal{C}}(S)|$.

In other words, a class $\mathcal{C}$ shatters a set of inputs S if any possible Boolean function on S can be represented by some $c \in \mathcal{C}$.

Definition *VCdim (Vapnik-Chervonenkis dimension)* of $\mathcal{C}$ is the maximum size of a set S shattered by $\mathcal{C}$:

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

If a maximum value does not exist, i.e., for every d there is a set S_d of size d which $\mathcal{C}$ shatters, then $VCdim(\mathcal{C}) = \infty$.

Result

Given this definition, we can show the following:

Theorem 3.2 *Let $\mathcal{H}$ be a concept class with VC-dimension d. $\mathcal{H}$ is PAC-learnable if and only if it has a finite VC-dimension.*

The VC-dimension also enters in the sample complexity bounds, and more, as discussed it class.

3.1.3 Examples

Linear halfspaces in the plane

Consider a real line in the plane. For $\mathbf{w} = (\alpha_1, \alpha_2, \theta), \mathbf{x} \in \mathbb{R}^2$, let

$$c_{\mathbf{w}}(\mathbf{x}) = 1 \iff \alpha_1 x_1 + \alpha_2 x_2 \geq \theta$$

All the positive points are above or on the line, and all the negative points are below the line. Define:

$$\mathcal{H} = \{c_{\mathbf{w}} | \mathbf{w} = (\alpha_1, \alpha_2, \theta) \in \mathbb{R}^3\}$$

We shall prove that $VCdim(\mathcal{H}) = 3$. For this concept class, any three points that are not collinear (i.e., lying on a single line) can be shattered. Figure 3.1(a) shows how one assignment out of the possible 8 assignments can be satisfied by a halfspace. To see that no set of four points can be shattered, we consider two cases. In the first case (shown in Figure 3.1(b)), all four points lie on the convex hull defined by the four points. In this case, if we label one ”diagonal” pair positive and the other ”diagonal” pair negative as shown in Figure 3.1(b), no halfspace satisfies this assignment. In the second case (shown in Figure 3.1(c)), three of the four points define the convex hull of the four points, and if we label the interior point negative and the hull points positive, again no halfspace can satisfy the labeling. Thus the VC–dimension here is three. In general, for halfspaces in $\mathbb{R}^d$, the VC–dimension is $d + 1$, as we have seen in class.

The Boolean parity function

Let $\mathcal{X} = \{0, 1\}^n$. Define the hypothesis h_T as

$$h_T(\mathbf{x}) = \bigoplus_{i \in T} x_i$$

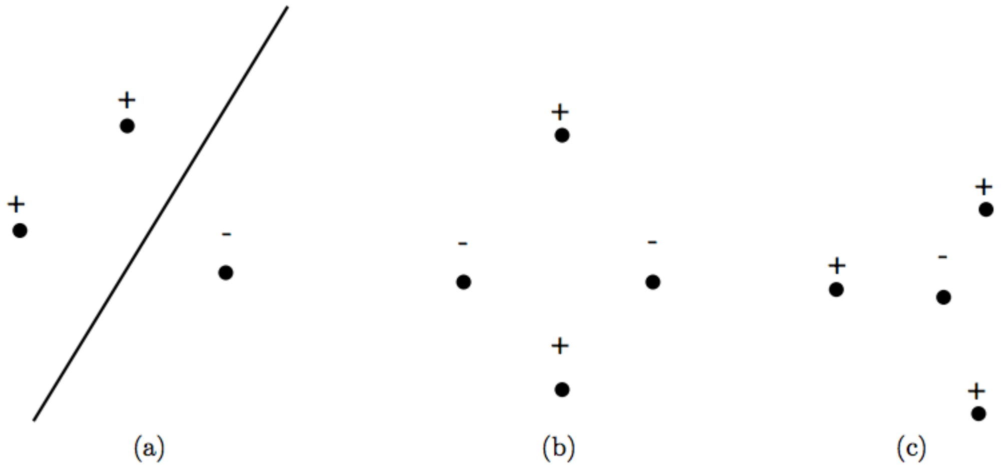

Figure 3.1: (a) An example of 3 non-collinear points and a halfspace that satisfies them; (b) and (c) are examples of impossible assignments.

where $T \subseteq \{1, ..., n\}$, and

$$\mathcal{H} = \{h_T | T \subseteq \{1, ..., n\}\}$$

We will first prove a lower bound: $VCdim(\mathcal{H}) \geq n$. Let $\mathbf{e}_i = \langle 0, \ldots, 0, 1, 0, \ldots, 0 \rangle$ be a unit vector, where '1' appears in the i-th place. We will show that the set $S = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ can be shattered by $\mathcal{H}$. To see that, we need to show that every label assignment for S is consistent with some hypothesis in $\mathcal{H}$. Let $y_1, ..., y_n$ be an assignment for the vectors $\mathbf{e}_1, ..., \mathbf{e}_n$. Then, choose the set

$$T = \{j : y_j = 1\}$$

We get

$$h_T(\mathbf{e}_i) = \begin{cases} 1 & \text{i} \in T \\ 0 & \text{i} \notin T \end{cases}$$

Thus, we conclude $\mathcal{H} \geq n$. We now show the upper bound: $\mathcal{H} \leq n$. We present two simple proofs for the upper bound:

1. There are 2^n parity functions. Thus $VCdim(\mathcal{H}) \leq \log_2 |\mathcal{H}| = \log_2 2^n = n$.

2. Given $n + 1$ vectors, there is a vector that is the linear combination of the others. Without loss of generality, assume it is the last one:

   $$\mathbf{x}_{n+1} = \mathbf{x}_1 \oplus ... \oplus \mathbf{x}_n$$

   Then, the values of $\mathbf{x}_1, ..., \mathbf{x}_n$ fix the value of $\mathbf{x}_{n+1}$, so for example, the assignment

   $$y_1 = 0, ..., y_n = 0, \; y_{n+1} = 1$$

   is impossible. Therefore, no set of size $n + 1$ can be shattered by $\mathcal{H}$.

3.2 Overfitting and Cross-Validation

Overfitting is a big problem. When we are learning the parameters of some classifier or a regression function, there is a danger that this classifier will not generalize from the data on which we've learned it, to new data we haven't yet seen.

For example, suppose we are given a set of points x_i, t_i, and we wish to fit a function $f(x) = t$ (e.g., Figure 3.2). We have as an input a set of examples S. We have an algorithm that given a sample S generates a hypothesis h_S (a suggested classifier or a regression function). We also have a loss function L, from which we can calculate the error of h_S's predictions. For example, the algorithm fits the best polynomial curve of a given degree M, and the loss function is the quadratic loss. It is evident that while the training error (i.e. the error calculated on the given set) decreases with M, the true error decreases and then increases, when overfitting becomes dominant (Figure 3.3).

Cross validation is a method to test the performance of your classifier when there is a limited amount of data available. We would like to get a better estimation of the true risk of the output hypothesis of a learning algorithm. An accurate estimation of the true risk can be obtained by using some of the training data as a validation set, over which one can evaluate the success of the algorithms output hypothesis.

The simplest way to estimate the true error of an hypothesis h is by sampling an addiional set of examples, independent of the training set, and using the empirical error on this validation set as our estimator. Then, we will see that the empirical error on the new set decreases, when we are still learning the "signal", and then increases, as we start overfitting.

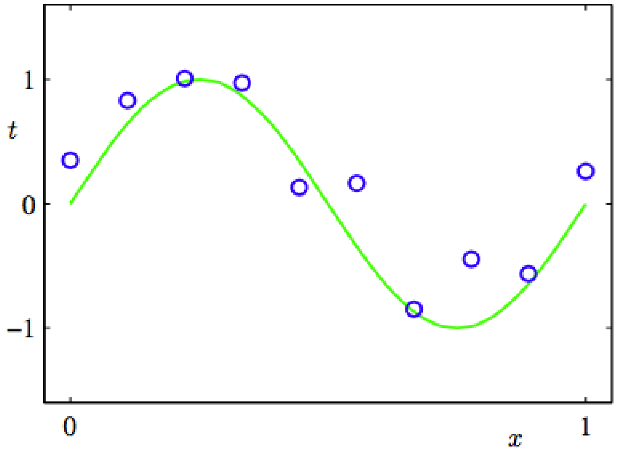

Figure 3.2: 10 pairs of (x_i, t_i), drawn from $t = sin(2\pi x)$ with random Gaussian noise. Adopted from *Christopher M. Bishop - Pattern Recognition and Machine Learning.*

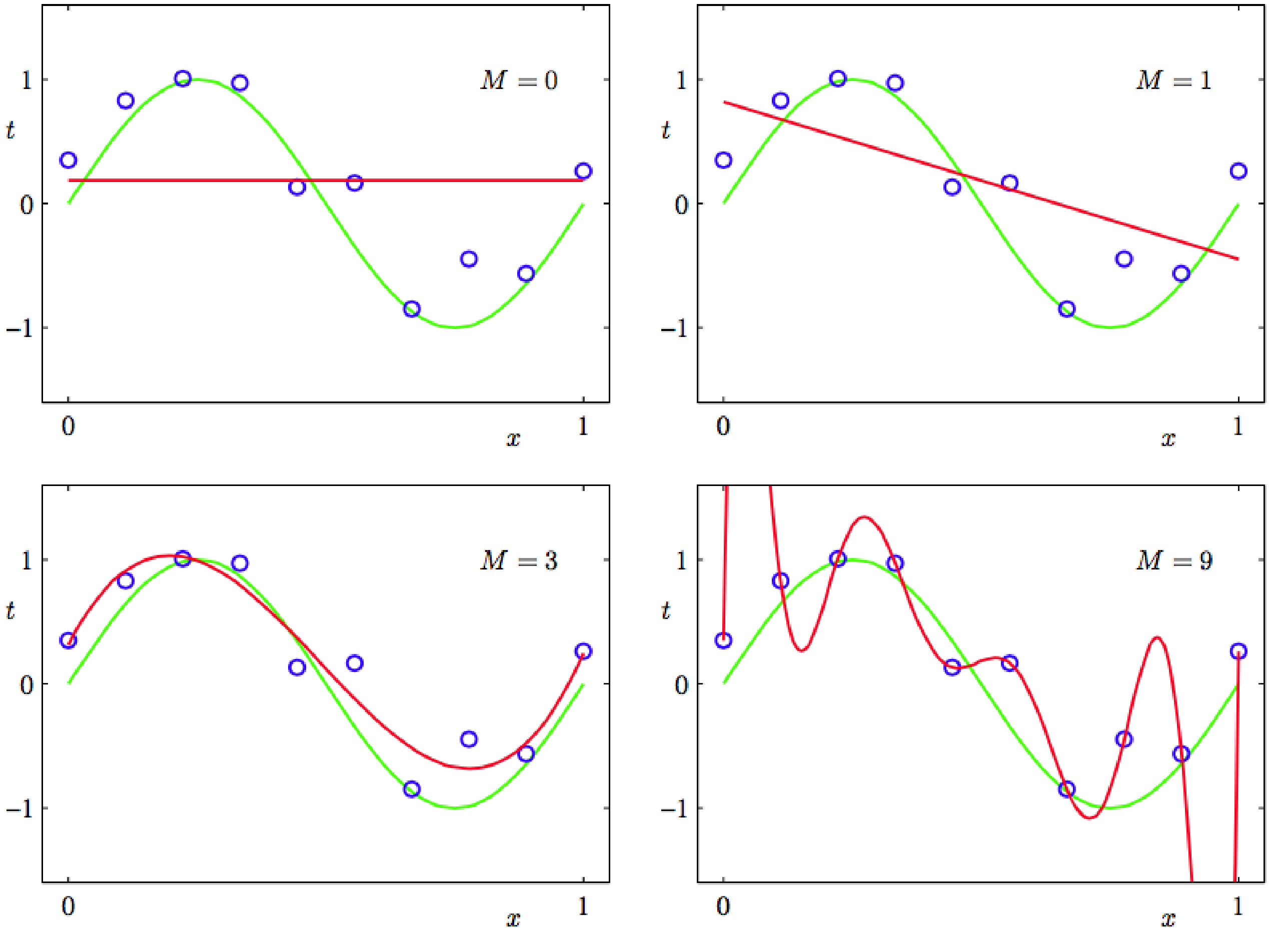

Figure 3.3: Different fitted polynomial curves for various degrees M.