Introduction to Machine Learning **Fall Semester**

Recitation 2

Lecturer: Regev Schweiger *Scribe: Regev Schweiger*

2.1 PAC - Review

Let the set X be the *instance space* or *sample space*. Let $\mathcal{D}$ be any fixed probability distribution over the space X. A *concept* or an *hypothesis* is a function from the sample space to a label: $c : X \to \{0,1\}$. A *concept class* over X is a set

$$\mathcal{C} \subseteq \left\{ c \mid c : X \to \{0,1\} \right\}.$$

Let c_t be the *target concept*, $c_t \in \mathcal{C}$. Let and $h \in \mathcal{H}$ be the *learned hypothesis*, where the *hypothesis class* $\mathcal{H}$. Generally, $\mathcal{H}$ may be a different concept class than $\mathcal{C}$. We will define the *error* of h with respect to the distribution $\mathcal{D}$ and the target concept c_t as follows:

$$error(h) = \Pr_{\mathcal{D}}[h(x) \neq c_t(x)] = \mathcal{D}(h \Delta c_t(x))$$

Let $EX(c_t, \mathcal{D})$ be a procedure (we will sometimes call it an *oracle*) that runs in a unit time, and on each call returns a labeled example $\langle x, c_t(x) \rangle$, where x is drawn independently from $\mathcal{D}$. In the PAC model, the oracle is the *only* source of examples for the learning algorithm.

Definition Let $\mathcal{C}$ and $\mathcal{H}$ be concept classes over X. We say that $\mathcal{C}$ is *PAC learnable by* $\mathcal{H}$ if there exists an algorithm A with the following property: for every concept $c_t \in \mathcal{C}$, for every distribution $\mathcal{D}$ on X, and for all $0 < \varepsilon, \delta < \frac{1}{2}$, if A is given access to $EX(c_t, \mathcal{D})$ and inputs ε and δ, then with probability at least $1 - \delta$, A outputs a hypothesis concept $h \in \mathcal{H}$: If $c_t \in \mathcal{H}$ (Realizable case), then h is satisfying

$$error(h) \leq \varepsilon$$

If $c_t \notin \mathcal{H}$ (Non-realizable), then h is satisfying

$$error(h) \leq \varepsilon + \min_{h' \in \mathcal{H}} error(h')$$

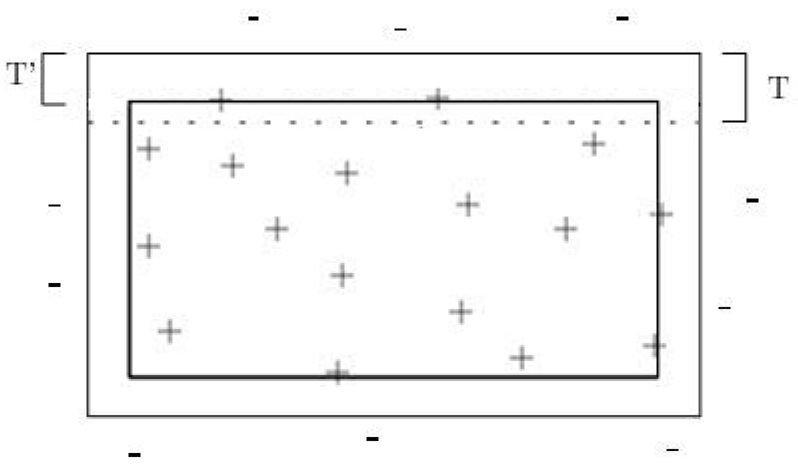

Figure 2.1: Adjusting strip size to have a weight of at most ε according to the real target function R.

2.2 Learning Rectangles

We give a proof that the concept class of rectangles discussed in class is PAC-learnable, and analyze the sample size.

In this section, we try to find what is a "sufficiently large" number of examples that is needed to learn a good hypothesis. For that, we will fix our accuracy and confidence parameters (ε, δ), and the strategy A. We will show that for any distribution D, we can assert sample size m that with high confidence (that is, probability at least $1 - \delta$), the returned rectangle R' from strategy A has an error of at most ε.

First, we will describe the strategy A: Given a sample S of size m, find the tightest fit rectangle - that is, the smallest rectangle which is consistent with the given sample. Effectively, this is done by set the boundaries of the rectangle to be the higher and lowest x_1 and x_2 coordinates of any points - see Figure 2.1 for illustration.

It is easy to see that by construction, R' is contained in R – and therefore, it can only make one type of error: Labeling positive samples as negative. What kind of a sample would be bad for our a strategy? Intuitively, if the sample is too concentrated in the center of the true rectangle R, then R' would be too small, and the error, $D(R\Delta R')$, would be too large. Therefore, our task is to show what is a sample size m, such that with high probability, a sample of size m is not too concentrated – and thus, that the error is sufficiently small. An equivalent way of saying the same thing, is that we need to show what is a sample size m, such that the probability that a sample of size m will be too concentrated is small. We now formalize this within the PAC framework.

To show this, we will construct strips $T_1, .., T_4$ such that $\forall i \;\; D(T_i) = \frac{\varepsilon}{4}$. For example, imagine a strip that extends from the top of the R rectangle downwards. As we move downwards and increase the strip height, the probability of the being in strip under D increases. We define our strip to be exactly at size for which $D(T_i) = \frac{\varepsilon}{4}$. (Note: In some discrete distributions, such a strip may not exist, and we will define our strip to be exactly at the smallest size for which $D(T_i) \geq \frac{\varepsilon}{4}$. The proof then follows. In the following, we will assume for clarity of presentation that we can define strips so that $D(T_i) = \frac{\varepsilon}{4}$).

Can we be sure that such a strip exists? Is it possible that we could extend the strip all the way to the other side of the rectangle? If so, this means that $D(R) \leq \frac{\varepsilon}{4} < \varepsilon$ - so clearly the error is smaller than ε, as $D(R\Delta R') \leq D(R) < \varepsilon$.

Now we formalize by what we mean that the sample is "too concentrated". From the construction we can see that T_i is independent of the sample and of R'. Note that we cannot certainly find the T_i but we can be sure that such T_i exists. Let the strips T'_i be the strip from the respective edge of R to the first sample. We would want to have $\forall i\ T'_i \subseteq T_i$ - intuitively, this means that the sample is not too concentrated. If that is the case, we obtained our requirement since

$$error(R') = \mathcal{D}(R\Delta R') = \mathcal{D}(\cup T'_i) \leq \mathcal{D}(\cup T_i) \leq \sum_{i=1}^{4} D(T_i) \leq 4 \cdot \frac{\varepsilon}{4} = \varepsilon$$

where we used the union bound.

From the construction, if there is at least one sampled point that resides in T_i it implies that $T'_i \subseteq T_i$. This is true, since the rectangle from strategy A must include all sampled positive points in R. To achieve that we can ask: what is the probability of sampling a bad sample set, that is, what is the probability that we didn't receive points from the sample data that are located on the constructive strips, i.e., T_i. Formally,

$$\Pr[\text{error} > \varepsilon] \leq \Pr[\exists i = 1..4\ \forall \mathbf{x} \in S, \mathbf{x} \notin T_i]$$

We will now try to bound this probability. By definition of T_i,

$$\Pr[\mathbf{x} \notin T_i] = 1 - \frac{\varepsilon}{4}$$

because in order for a point $\mathbf{x}$ to be out of T_i, it has to be either negative (and thus is outside R) or positive (and thus inside of R but outside of T_i). Since our sample data is i.i.d from distribution D:

$$\Pr[\forall \mathbf{x} \in S, \mathbf{x} \notin T_1] \leq \left(1 - \frac{\varepsilon}{4}\right)^m$$

The same analysis holds on each T_i strip. Hence, by the union bound, we get that on the entire region the error would not exceed the sum of probabilities for each of the strips. That is,

$$\Pr[\text{error} > \varepsilon] \leq 4\left(1 - \frac{\varepsilon}{4}\right)^m$$

From the inequality $(1-x) \leq e^{-x}$, we obtain:

$$\Pr[\text{error} > \varepsilon] \leq 4\left(1-\frac{\varepsilon}{4}\right)^m \leq 4e^{-\frac{\varepsilon}{4}m} < \delta$$

That is, if we want to have accuracy ε and confidence of at least $1-\delta$, we have to choose the sample size m to satisfy:

$$4e^{-\frac{\varepsilon}{4}m} < \delta \Leftrightarrow m > \frac{4}{\varepsilon} ln \frac{4}{\delta}$$

For this strategy A, and for every small (ε, δ) we like, we got the sample size that is needed for having a good learner.

2.2.1 Remarks

1. The analysis holds for any fixed probability distribution $\mathcal{D}$, we only required that the sample points are i.i.d from distribution $\mathcal{D}$ to obtain our bound.

2. The minimal sample size $m(\varepsilon, \delta)$ behaves as we might expect. One might want to have better accuracy by decreasing ε or greater confidence by decreasing δ — our algorithm requires more examples to meet those requirements. There is a stronger dependence in ε.

3. The parameter ε gives the degree of accuracy that we want to achieve. It determines what is a good hypothesis for achieving a good approximation in respect to the target function. In our example, the accuracy determines which of our hypothesis rectangles are good enough in respect to the real rectangle target. We pay attention that the accuracy does not depend on the data distribution.

4. The parameter δ gives the degree of confidence on having a good learner. Meaning, how sure are we that we've reached that level of accuracy. This can be related on, how typical the given sample data reflects the true distribution. Again, it does not depend on the data distribution.

5. We might have cases as shown in Figure 2.2, where the distribution $\mathcal{D}$ gives large weights to particular regions of the plain, creating a distorted image of the rectangle. In any case, under those conditions, since the learner is tested on the same distribution $\mathcal{D}$, and this distribution has small error between R and R', the rectangle $\mathcal{R}'$ will be a good hypothesis (in respect to ε, δ).

6. The strategy A that we defined is efficient: In computational view, the only need is to search for the max and min points that defines our tightest-fit rectangle. In sample data size view, the number of examples that is required for achieving accuracy ε with confidence $1-\delta$ is polynomial in $\frac{1}{\varepsilon}$ and $ln\frac{1}{\delta}$.

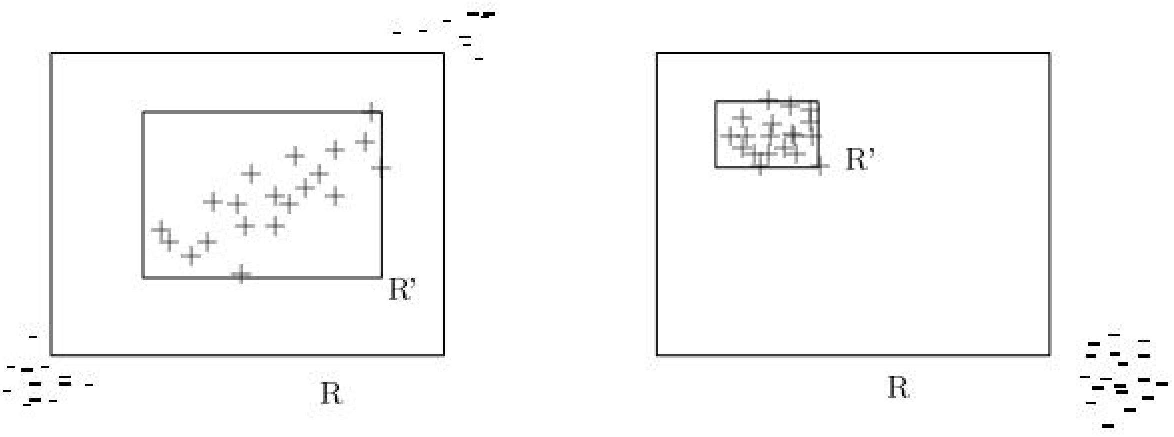

Figure 2.2: Two cases depending on the sample size

7. In this example, as opposed to the Bayesian approach, we haven't been trying to model $\mathcal{D}$ or to guess which rectangle is more likely (prior). We have separated the distribution $\mathcal{D}$ from the target function (rectangle R), and directly try to predict hypothesis for this function.

2.2.2 Finite Hypothesis Class Example - Majority Circuits

We will now give an example application of the PAC-learnability of finite hypothesis classes. First, we recall the result in the realizable case:

Theorem 2.1 *Every finite hypothesis class $\mathcal{H}$ is PAC-learnable, with sample complexity*

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

where $1-\delta$ is the confidence and ε is the error.

As an example, we apply this result to the case of majority circuits (or majority network):

Definition We define a *majority circuit* C with n inputs, D levels as follows. C is a circuit which encodes a function by a layer graph with D layers (levels), and n nodes in each layer. The graph C has n input nodes, which are nodes with no incoming edges, in which we assign the values of the input variables; and a single output node, with no outgoing edges, in which the resulting value will be assigned. Each node in each layer has incoming edges, which are the output edges of all the nodes in the previous layer (or the input variables, for the first layer). Therefore, each node has n incoming edges. For each such node $v_{i,j}$ (the i-th node in the j-th layer), each incoming edge $k = 1, \ldots, n$ has a weight $w_{i,j,k} \in \{-1, 0, 1\}$. The output of the node $v_{i,j}$ is the sign of the weighted sum of its entries:

$$output(v_{i,j}) = sign\left(\sum_{k=1}^{n} w_{i,j,k} \cdot input_k(v_{i,j})\right)$$

Let $\mathcal{H} = C_{n,D}$, where $C_{n,D}$ is the set of all functions that can be calculated by a majority circuit with n gates and D levels. How many elements exist in $\mathcal{H}$? Each node has n parameters to be determined, so the total number of such parameters is $n \cdot (n(D-1)+1)$. Each such parameter can be either $-1, 0$ or 1. So finally, we get $|\mathcal{H}| = 3^{n(n(D-1)+1)}$.

So, how many samples do we need to learn (according to the PAC framework) a majority circuit? According to the result above,

$$m(\varepsilon, \delta) \geq \frac{1}{\varepsilon} \log \frac{|\mathcal{H}|}{\delta} = \log 3 \cdot \frac{n(n(D-1)+1)}{\varepsilon} + \frac{1}{\varepsilon} \log \frac{1}{\delta}.$$