Introduction to Computational Learning Theory **Spring 2021**

Lecture 1: Introduction – Learning Rectangles

Lecturer: Roi Livni *Scribe:*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
... AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.
IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.
This xkcd comic was posted in 2014. Five years later we can safely say that we have achieved it.

Roughly, Machine Learning studies the ability of a computer to be "programmed by experience".

1. Classification.

    - Image Recognition
    - Character Recognition
    - Voice Recognition.
    - etc..

2. Reinforcement Learning
   - Control in an unknown enviroment.
   - Autonomous driving.
3. Unsupervised
   - Generative Learning.
   - Clustering.
   - etc..

1.1 Theoretical Machine Learning

Developing Basic models that we can analyze, understand and study the basic concepts and phenomenas that allow machines to learn.

Basic Questions

- How many examples a learner needs to observe inorder to learn a task?
- What types of algorithms are there?
- What are the computational limits for learning?
- What are the natural assumptions that allow learning?
- What prohibits learning?

We begin with some simple motivating examples.

- Suppose we are given examples of patients, some that have the disease and some that don't.
  - Each patient is described by a list of features and results of tests (e.g., the patient's age, weight, blood pressure, etc.).
  - For each patient we are told whether he/she has the disease or not.

  - Given these patients, we want to come up with a rule by which we can diagnose new patients well.

- As another motivating problem, consider the task of handwriting recognition. Here we are given examples of handwritten digits, and with each we get a label of what digit it is. Based on these examples we would like to come up with a rule by which we can label new examples of handwritten letters that we haven't yet seen.

- These examples motivate studying learning algorithm that are given as input *labeled examples* (patient,sick/not sick), (digit,0/1/..), and in turn, the algorithm output a *prediction rule.*

- This rule is then used to label new, yet unseen examples.

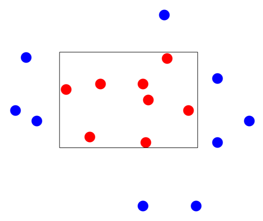
1.2 Learning (axis-aligned) rectangles

We start then with a simplified example that very roughly captures the above task

- Suppose we want to come up with a rule for the concept of "medium built". That is, our examples are points (x, y) in the plane, where x is the weight of a person and y is the height. We assume for simplicity that these values are normalized so that $(x, y) \in [0, 1]^2$.

- Then a reasonable assumption is that the concept of "medium built" can be described by an axis-aligned rectangle R in $[0, 1]^2$, that is, by four values $\ell_x < h_x, \ell_y < h_y$, where for every (x, y) in the rectangle, the label is positive ('+') and for every (x, y) outside the rectangle, the label is negative ('−'). In other words, the concept is defined by a function f_R, where $f_R(x, y) = +$ if $\ell_x \leq x \leq h_x$ and $\ell_y \leq y \leq h_y$, and $f_R(x, y) = -$ otherwise.

- Suppose that someone labels some examples as positive or negative (that is, the person can label correctly who has medium built and who hasn't though he doesn't have a rule he can give us).



- A natural idea is, given the labeled examples, to find a rectangle that is *consistent* with the examples. That is, a rectangle, $\widehat{R}$ that contains all positive examples and doesn't contain any negative example.

- Concerelly, we may take $\widehat{R}$ to be the *minimal* such consistent rectangle. That is, $\widehat{R}$ is defined by $\hat{\ell}_x, \hat{h}_x, \hat{\ell}_y, \hat{h}_y$ where $\hat{\ell}_x$ is the minimum value of the x coordinate, taken over all positive examples in the sample, and $\hat{h}_x, \hat{\ell}_y, \hat{h}_y$ are defined analogously.

- Do we get a good hypothesis?

- First, we need to define a notion of "goodness".

Set $\Delta(R, \widehat{R})$ to be the region on which two rectangles disagree (their symmetric difference).

$$\Delta(R, \widehat{R}) := \{(x, y) : \ f_R(x, y) \neq f_{\widehat{R}}(x, y)\}.$$

Denote A reasonable definition of a good hypothesis, is that $\Delta(R, \widehat{R})$ should be "small".

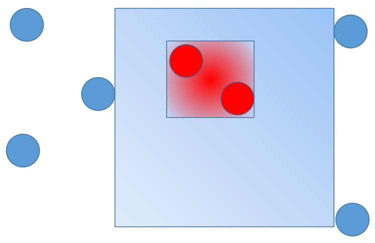


Figure 1.1: An illustration for a bad (adversarial) choice of a sample.

- **Approximation error:** A crucial observation, though, is that we care about the *error* rate. In particular, if (x, y) are i.i.d distributed according to distribution D, then a relevant notion of "size" is $D(\Delta(R, \widehat{R}))$.

  More accurately, the prediction rule we suggested will have error rate ϵ (i.e. the probability of false prediction is ϵ) if for $\hat{R}$ we have:

$$D(\Delta(\hat{R}, R)\}) < \epsilon.$$

Correct in Probability: Note that $\hat{R}$ is also a random variable that depends on the observed examples. It might be unrealistic to expect that our algorithm will work well for every observed sample. Imagine, that the algorithm never observed negative examples, this event will cause our algorithm to fail, however this event is also highly unlikely (if we saw "enough examples"). Thus, we might want our algorithm to succeed w.h.p. Denoting by P the probability over the IID sequence of observed examples, then we could say that the algorithm is successful if

$$\mathbb{P}\left(D\left(\Delta(\hat{R}, R)\right) > \epsilon\right) < \delta.$$

- Here $\hat{R}$ is a random variable that depends on the observed sample:

  D and R are assumed to be fixed but unknown!.

Let us now analyze our suggested algorithm. Suppose that we have observed a sample of size

$$m \geq (4/\epsilon)\ln(4/\delta).$$

We will show that with probability at least $1-\delta$ (over the choice of the sample), we have that

$$D(\Delta(R, \widehat{R})) \leq \epsilon.$$

- Observe first that if the size of R (i.e. $D(R)$) is at most ϵ, then no matter what sample points we get (and in particular, we may get only negative examples), $\Delta(R, \widehat{R}) \leq \epsilon$.
- Therefore, assume that the size of R is at least ϵ.
- We next define 4 sub-rectangles within R.
- Each sub-rectangle A_i shares three of the bounds ℓ_x, h_x, ℓ_y, h_y and differs in one, and the area of each sub-rectangle is exactly $\epsilon/4$ (i.e. $D(A_i) < \epsilon/4$).
- Note that these sub-rectangle are independent of our algorithms choice!!!

- The main simple observation is that if the sample includes at least one sample point from each sub-rectangle, then $\Delta(R, \widehat{R})$ is bounded by the area of the union of these sub-rectangles, which is less than $4 \cdot (\epsilon/4) = \epsilon$.
- Therefore, it remains to show that with probability at least $1 - \delta$, the sample will indeed include at least one point from each sub-rectangle.
- Equivalently, we want to show that the probability that for one of these sub-rectangles, no sample point falls in the sub-rectangle, is at most δ. To this end we defined four events: E_1, E_2, E_3, E_4, where E_i is the ("bad") event that no sample point fell into A_i.

Consider any particular E_i. Since the probability that a single sample point falls into A_i is $\epsilon/4$, the probability that a single sample point does not fall into A_i is $(1-\epsilon/4)$. Since the sample points are selected independently, we get that for each i:

$$\mathbb{P}[E_i] = (1 - \epsilon/4)^m < e^{-(\epsilon/4)m} = e^{-(\epsilon/4)\cdot(4/\epsilon)\log(4/\delta)} = \delta/4 \ .$$

We now want to bound $\mathbb{P}(\cup_{i=1}^4 E_i)$

The Union Bound: Let $F_1, F_2, \ldots, F_m$ be any arbitrary events. Then:

$$\mathbb{P}[\cup_{i=1}^m F_i] \leq \sum_{i=1}^m \mathbb{P}[F_i]$$

Applying union bound for our case: since the probability that for *some* sub-rectangle, no sample point fell into the sub-rectangle is $\mathbb{P}\left[\bigcup_{i=1}^4 E_i\right]$, if we apply the union bound we get that:

$$\mathbb{P}[\Delta(R, \widehat{R}) > \epsilon] \leq \mathbb{P}\left[\bigcup_{i=1}^4 E_i\right] \leq \sum_{i=1}^4 \mathbb{P}[E_i] \leq \delta \ ,$$

and we have established what we wanted.