1 Lecture 1: 2017.01.23

1.1 Course Outline/Description

1.1.1 Formal Models of ML

- Assumptions about data generation process.
- Assumptions about what algorithms ”knows”.
- Sources of info that the algorithms has
- Criteria/objective of learning is.
- Restrictions on algorithms.

1.1.2 Examples of Models

- ”PAC” model (in first 1/2 of the term).
- Statistical learning theory.
- ”no-regrets” learning models.
- Reinforcement learning.
- ML & Differential privacy.
- ”Fairness” in ML

1.2 A Rectangle Learning Problem

Suppose you are trying to teach an alien friend the ”shape” of humans in terms of abstract descriptions like ”medium build”, ”athletic”, etc. We are going to assume that each one of these descriptions represents a rectangular region on the height-weight plane but we are not aware of the exact dimensions. The only thing we are able to tell the alien is whether a particular individual is medium built or not. i.e. we can only label examples.

- target rectangle R, the *true* notion of ”medium built”.
- hypothesis rectangle $\hat{R}$.

Remark 1.1. *Note that the assumption that the classifier function is a rectangle is rather strong. There is always this trade-off to be able to actually compute things. From a Bayesian point of view, “we always need a prior”.*

Given a data cloud of examples, a reasonable hypothesis rectangle could be the tightest fit rectangle. However, this choice ignores the negative examples so it seems that that we are throwing away information. In a sense, this rectangle would be the least likely.

- Assume $\langle x_1, x_2 \rangle$ pairs of height-weight are i.i.d. from an arbitrary unknown probability distribution, D.

We want to be able to evaluate how our hypothesis rectangle is performing. We want bounds on the classification error, which can be thought as the size of the symmetric difference between R and $\hat{R}$

$$D[R \triangle \hat{R}] \equiv \mathbb{P}_D[R(\vec{x}) \neq \hat{R}(\vec{x})]. \tag{1.1}$$

Theorem 1.2. *Given $\varepsilon, \delta > 0$, there is some integer N such that if we have more than N training examples,*[1] *we have*

$$D[R \triangle \hat{R}] \equiv \mathbb{P}_D[R(\vec{x}) \neq \hat{R}(\vec{x})] < \varepsilon, \tag{1.2}$$

with probability at least $1 - \delta$.

Proof. First of all, note that using tightest fit, the hypothesis rectangle is always contained inside the target rectangle. Now, for each side of the target rectangle we may draw inward strips in such a way that each strip has $\mathbb{P}_D[Strip] < \frac{\varepsilon}{4}$. If the training set has a positive example in each of these four strips, then the inequality above is satisfied because the boundary of the hypothesis rectangle would be contained in the union of the strips. Next we need to deal with the required sample size to obtain this result with some certainty. Let m denote the sample size, since the distribution is i.i.d., we have

$$\begin{aligned} \mathbb{P}_D[\text{miss a specific strip m times}] &= \left(1 - \frac{\varepsilon}{4}\right)^m, \\ \mathbb{P}_D[\text{miss any of the strips m times}] &\geq 4\left(1 - \frac{\varepsilon}{4}\right)^m. \end{aligned}$$

By the discussion above, the last inequality implies

$$\mathbb{P}_{D^m}[D[R \triangle \hat{R}] \geq \varepsilon] \leq 4\left(1 - \frac{\varepsilon}{4}\right)^m,$$

which can be chosen arbitrarily small for big enough m. One can obtain $N \geq \frac{4}{\varepsilon} \ln\left(\frac{4}{\delta}\right)$. □

Remark 1.3. *This proof generalizes to d-dimensional rectangles. We only need to replace 4 with $2d$, the number of $(d-1)$-faces. We can also try to incorporate noisy data, where the labels have some probability of being wrong.*

1.3 A More General Model

- Input/instance/feature space, X. (e.g. $\mathbb{R}^2$ in the example above)
- Concept/classifier/boolean function, $C : X \to \{0, 1\}$ or we can also think about it as an indicator function of the positive examples or the subset of positive examples.

[1] This the same as saying that sample size is at least N.

- Concept class/target class, $\mathcal{C} \subset \mathcal{P}(X)$, the admissible concepts/classifiers. (e.g. all rectangles in $\mathbb{R}^2$ in the example above)

- target concept, $c \in \mathcal{C}$. (e.g. target rectangle in the example above)

- Input distribution, D over X (arbitrary & unknown)

- Learning algorithm given access to examples of the form: $\langle \vec{x}, y \rangle$ where $\vec{x}$ is i.i.d. drawn from D and $c(\vec{x}) = y$.

Definition 1.4 (PAC Learning)**.** We say that a class of functions over X, $\mathcal{C}$, is *Probably Approximately Correct (PAC) learnable* if there exists an algorithm, L, such that given any c in $\mathcal{C}$, D a distribution over X and $\varepsilon, \delta > 0$, L with these parameters and random inputs $\vec{x}$'s satisfies:

- (Learning) With probability $\geq 1-\delta$, L outputs a hypothesis, h in $\mathcal{C}$ such that $D[h \triangle c] < \varepsilon$, i. e.

$$Err(h) := \mathbb{P}_{x \sim D}[h(x) \neq c(x)] \leq \varepsilon. \tag{1.3}$$

- (Efficient) L runs in time/sample $poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \text{dimension}\right)$.