Computational Learning Theory Lecture Notes

Martin Citoler-Saumell

CIS 625 Spring 2017

- Instructor: Prof. Michael Kearns.

1 Lecture 1: 2017.01.23

1.1 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 $\hat{R}$

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

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*

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

with probability at least $1 - \delta$.

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

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

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

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.
- 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.

$$Err(h) := \mathbb{P}_{x \sim D}[h(x) \neq c(x)] \leq \varepsilon. \tag{1}$$

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

2 Lecture 2: 2017.01.30

Remark 2.1 (PAC Learning)**.** *Fixing the class $\mathcal{C}$ is a strong assumption, it is the prior you are assuming about the true behavior of the data. For example, when you fit a linear regression, you are assuming that there is a true linear relation in the data.*

In contrast, the assumption on the distribution over X is fairly general.

Theorem 2.2. *The class of rectangles over $\mathbb{R}^2$ from example above is PAC learnable (with sample size $m \sim \frac{1}{\varepsilon} \ln\left(\frac{1}{\delta}\right)$).*

2.1 PAC learning boolean conjunctions

In the following we are going to see an example of problem that is PAC learnable.

- $X = \{0, 1\}^n$
- Class $\mathcal{C}$ = all conjunctions over $x_1, \ldots, x_n$. $|\mathcal{C}| = 3^n$
  E.g.: If $c = x_1 \wedge \sim x_3 \wedge \sim x_{11} \wedge x_{26} \ldots$,

$$c(\vec{x}) = 1 \iff x_1 = 1, x_3 = 0, x_{11} = 0, x_{26} = 1, \ldots \tag{2}$$

- D over $\{0, 1\}^n$.

2.1.1 Algorithm for monotone case (i.e. no $\sim$'s)

In this case we are trying to fit something like $c = x_1 \wedge x_5 \wedge x_{13} \ldots$. i.e. given some examples of sequences of bits and the result of c on them, we are trying to guess what the conjunction c actually is. We can use the following algorithm:

- $h \leftarrow x_1 \wedge x_2 \wedge x_3 \wedge \ldots \wedge x_n$, start with the conjunction of all the variables.
- For each positive example, $\langle \vec{x}, 1 \rangle$, delete any variable in h such that $x_i = 0$.
  This method ensures that the positives of h is a subset of the true c.

Analysis

Let p_i denote the probability we delete x_i from h in a single draw. In other words,

$$p_i = \mathbb{P}_{\bar{X} \sim D}[c(\vec{x}) = 1, x_i = 0]. \tag{3}$$

Then we have an a priori bound on error: $Err(h) \leq \sum_{x_i \in h} p_i$. We can make a distinction between *bad* and *good* indices. An index, i, is bad if $p_i \geq \frac{\varepsilon}{n}$ and good otherwise. Note that if h contains no bad indices, then we have

$$Err(h) \leq \sum_{x_i \in h} p_i \leq n \left(\frac{\varepsilon}{n} \right). \tag{4}$$

Let's fix some index, i, such that $p_i \geq \frac{\varepsilon}{n}$. i.e. a bad index. We have that

$$\mathbb{P}[x_i \text{ "survives" m random samples}] = (1 - p_i)^m \quad (iid) \tag{5}$$

$$\leq \left(1 - \frac{\varepsilon}{n} \right)^m \tag{6}$$

$$\mathbb{P}[\textit{any} \text{ bad i "survives" m random samples}] \leq n \left(1 - \frac{\varepsilon}{n} \right)^m. \tag{7}$$

If we want the right-hand-side of the last inequality to be less than some $\delta > 0$, we end up with $m \geq \frac{n}{\varepsilon} \ln\left(\frac{n}{\delta}\right)$. In other words, we just proved the following theorem

Theorem 2.3. *Conjunctions over $\{0,1\}^n$ are PAC learnable with sample size $m \geq \frac{n}{\varepsilon} \ln\left(\frac{n}{\delta}\right)$.*

Remark 2.4. *An analogous argument proves this theorem for conjunctions that are not necessarily monotone. The only difference is that we have to keep track the extra $\sim$ variables.*

Remark 2.5. *We can identify some pattern for this kind of analysis. We identify some "bad" things that may happen and then prove that the probability of them happening decreases fast when we increase the number of samples seen.*