CS 395T Computational Learning Theory

Lecture 2: September 12, 2005 Lecturer: Adam Klivans

Scribe: Alex Sherstov

PAC Learning

This lecture describes the model of *probably approximately correct* (PAC) learning, introduced by Valiant in 1984. The model is illustrated with learning algorithms for two concept classes: axis-aligned rectangles and Boolean disjunctions.

2.1 Definitions

The *input space* X is the set of all instances of interest; we will typically work with the Boolean input space $X = \{0,1\}^n$. A *concept* $c : X \to \{0,1\}$ is a Boolean function defined over the input space. A *concept class* $\mathcal{C} = \{c_1, c_2, \dots\}$ is a set of concepts.

In PAC learning, the objective is to learn a fixed but unknown concept $c \in \mathcal{C}$ with respect to a fixed distribution D over the input space. This means that the learner receives a sequence of examples $(x_1, c(x_1)), (x_2, c(x_2)), \dots$, drawn independently at random from the distribution D and labeled by the target concept c. The goal of the learner is to output a hypothesis $h \in \mathcal{C}$ that is ϵ-accurate w.r.t. c over D: $\Pr_{x \sim D}[h(x) \neq c(x)] < \epsilon$.

A concept class $\mathcal{C}$ is *PAC learnable* if there is an algorithm L such that for every concept $c \in \mathcal{C}$ and every choice of δ, ϵ with $0 < \delta, \epsilon \leq 1/2$, with probability at least $1 - \delta$ algorithm L outputs a hypothesis $h \in \mathcal{C}$ satisfying $\Pr_{x \sim D}[h(x) \neq c(x)] < \epsilon$.

An algorithm *runs in time* t if it draws at most t examples and requires at most t time steps. A concept class is *efficiently PAC learnable* if it is PAC learnable by an algorithm that runs in time polynomial in $1/\epsilon, 1/\delta$, and instance length.

2.2 Example: Axis-Aligned Rectangles

Consider the input space $X = \mathbb{R}^2$ of two-dimensional points and the concept class $\mathcal{C}$ of axis-aligned rectangles. Consider the following algorithm for $\mathcal{C}$: given a set of examples, output the tightest rectangle that encompasses the positive points. Observe that this algorithm always outputs a consistent hypothesis, namely, a rectangle contained inside the concept (and thus correct on all negative instances).

Given m examples, what is the probability δ that the algorithm outputs a hypothesis with error ϵ or more? Consider any hypothesis (rectangle) h with such error. Consider the region outside h but inside the target concept; its weight w.r.t. to the given distribution D is ϵ, by assumption, since h is

correct on all negative instances. Then one of the four bands of that region (2 vertical, 2 horizontal) must have weight at least $\epsilon/4$ w.r.t. D. The likelihood of this event given m examples is at most $4(1-\epsilon/4)^m$. This is our δ. Solving for m yields

$$m \geq \frac{4}{\epsilon} \ln \frac{4}{\delta}.$$

This analysis shows that the proposed algorithm is indeed a PAC algorithm for $\mathcal{C}$.

2.3 Another Example: Disjunctions

Our second example is the class of disjunctions on Boolean literals $\{x_1, x_2, \ldots, x_n\}$. Consider the familiar elimination algorithm: start with all literals in the disjunction and, on each negative example, eliminate the active literals.

Fix a concept c. What is the probability δ that after m examples, the algorithm will produce a hypothesis with error ϵ or worse w.r.t. c? For each literal y, define $p(y)$ to be the fraction of inputs $x \in \{0,1\}$ that activate y but make c false: on such inputs, having y in the disjunction will contribute at most $p(y)$ to the error. If $p(y) < \epsilon/n$ for all y, the initial hypothesis is ϵ-accurate. If there is some y with $p(y) \geq \epsilon/n$, the probability that y will remain in the disjunction after m examples is at most $(1-\epsilon/n)^m$. By the union bound, the probability that *any* such literal y will remain in the disjunction after m examples, is at most $n(1-\epsilon/n)^m$. This is our δ. Solving for m yields

$$m \geq \frac{n}{\epsilon} \ln \frac{n}{\delta},$$

proving that the proposed algorithm is a PAC learner. The reader is invited to extend the analysis to the case when the concept class additionally contains disjunctions with *negated* variables.