THEORETICAL MACHINE LEARNING COS 511

LECTURE #15 MARCH 29, 2016

LECTURER: ELAD HAZAN SCRIBE: CARSON EISENACH

1 Weak Learnability and Boosting

1.1 Strong learnability vs Weak learnability

Recall the definition of a PAC-learning problem $(\mathcal{X}, \mathcal{Y}, \mathcal{H}, \ell)$ from the first few lectures. We define here a new notion of learnability, called *weak learnability*. (For simplicity, the following definition makes the realizability assumption and uses the $0/1$ loss.)

Definition 1.1. A learning problem is *weakly learnable* if there exists $\gamma > 0, m : (\delta, \gamma) \to \mathbb{N}$ and an efficient algorithm such that for all $\delta \in (0, 1]$, the algorithm can, after seeing $m(\delta, \gamma)$ examples from distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, return $h \in \mathcal{H}$ such that with probability $1 - \delta$

$$\text{error}(h) \leq \frac{1}{2} - \gamma$$

So now we have two types of learnability: PAC learnability (also called *strong learnability*) and weak learnability. A natural question is are these two equivalent? It is clear that strong learnability implies weak learnability (the two definitions differ only in the error bound), but the converse is not at all obvious. We show today that (surprisingly) the converse is in fact true, and thus weak learnability is (in some sense) equivalent to strong learnability.

1.2 A simple boosting algorithm

"Boosting" is an intuitive approach to this: we construct a strong learning algorithm by taking a majority vote of many weak learning algorithms. Algorithm 1 below gives a high level picture, while Algorithm 2 gives specific details of how boosting works.

```
Algorithm 1 High-level Idea for Generic Boosting Algorithm
Input: A weak learner $\mathcal{A}$ such that over $m$ samples according to distribution $p$, the hypothesis $\mathcal{A}(p) \in \mathcal{H}$
    satisfies error($\mathcal{A}(p)$) $\leq \frac{1}{2} - \gamma$ with probability $1 - \delta$
Output: $\hat{h} \in \mathcal{H}$ such that error($\hat{h}$) $\leq \varepsilon$ with probability $1 - \delta$
```

```
Algorithm 2 A Simple Boosting Algorithm
Input: A weak learner $\mathcal{A}$, a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, number of iterations $T$, and parameters $\eta, \delta, \varepsilon > 0$.
Output: Hypothesis $h$ such that on the sample $S$, error($h$) $= 0$.
Take $m = \frac{|\mathcal{H}|}{\varepsilon^2} \log \frac{1}{\delta}$ samples from $\mathcal{D}$. Denote this set by $S$.
Define a distribution $p_t \in \Delta_m$ over $S$. Set $p_1 = \frac{1}{m} j$.
for $t = 1$ to $T$ do
    $h_t \leftarrow \mathcal{A}(p_t)$
    Update $\tilde{p}_{t+1}(i) \leftarrow p_t(i) \exp(-\eta r_t(i))$ where $r_t(i) := 1\{h_t(x_i) = y_i\}$
    $p_{t+1} \leftarrow \frac{\tilde{p}_{t+1}}{|\tilde{p}_{t+1}|}$
end for
return $\overline{h}_T = maj(h_1, \ldots, h_T)$
```

The following theorem analyzes the performance of Algorithm 2. (In what follows, we use the notation error_S to denote the empirical error on the sample S.)

Theorem 1.2. *After* $T = \frac{\log m}{\gamma^2}$ *iterations,* $\mathrm{error}_S(\overline{h}_T) = 0$ *with probability* $1 - \delta$.

Proof. Assume that the weak learner $\mathcal{A}$ always succeeds (this technically occurs with probability at least $1 - \delta$; see Professor Hazan's notes for full details). This means that $\mathrm{error}_{p_t}(h_t) \leq \frac{1}{2} - \gamma$ for all t. Because of how we defined r_t, $p_t^T r_t = 1 - err_{p_t}(h_t)$. Thus:

$$\frac{1}{2} + \gamma \leq \frac{1}{T} \sum_{t=1}^{T} p_t^T r_t$$

Furthermore by the regret bound on the MW algorithm, which we can apply since Algorithm 2 performs MW updates,

$$\frac{1}{T} \sum_{t=1}^{T} p_t^T r_t \leq \min_{p^* \in \Delta_m} \frac{1}{T} \sum_{t=1}^{T} (p^*)^T r_t + \sqrt{\frac{\log m}{T}}$$

Assume for sake of contradiction that $\mathrm{error}_S(\overline{h}_T) \neq 0$. Then there exists some example $i^* \in S$ on which more than half of the h_t err. This implies that

$$\min_{p^* \in \Delta_m} \frac{1}{T} \sum_{t=1}^{T} (p^*)^T r_t + \sqrt{\frac{\log m}{T}} \leq \frac{1}{T} \sum_{t=1}^{T} r_t(i^*) + \sqrt{\frac{\log m}{T}} < \frac{1}{2} + \gamma$$

where the middle equation is due to a simple averaging argument, and the last equality follows by our choice of T. This gives a contradiction, and thus $\mathrm{error}_S(\overline{h}_T) = 0$. □

In words, we have just shown that Algorithm 2 finds a hypothesis that gives 0 error on the sample. From standard PAC-learning, we can find a large enough sample size such that whenever a hypothesis gives 0 error on a sample of that size, then with probability at least $1 - \delta$ it has at most ε error on the actual distribution (i.e. it "generalizes well").

There is, however, a slight nuance: the hypothesis that Algorithm 1 chooses is of course not guaranteed to be in $\mathcal{H}$. Instead, it comes from the set of hypotheses which can be written as the majority of T hypothesis in $\mathcal{H}$. Let us call this set $\overline{\mathcal{H}}_T$.

Since T is a function of m, and m is a function of $|\overline{\mathcal{H}}_T|$, thus T depends on the dimension of $|\overline{\mathcal{H}}_T|$. But this is problematic since $|\overline{\mathcal{H}}_T|$ also depends on T. To fix this, note that in the finite case, $dim(\overline{\mathcal{H}}_T) \leq T dim(\mathcal{H})$, yielding an equation we can solve for T. It can be shown that

$$T \propto \frac{dim(\mathcal{H}) \log(dim(\mathcal{H}))}{\varepsilon^2} \log(\frac{1}{\delta})$$

is sufficient for the algorithm to generalize well (i.e. within ε error on the actual distribution with probability at least $1 - \delta$).

1.3 AdaBoost

An especially popular boosting algorithm is AdaBoost. The only difference between it and Algorithm 2 is that AdaBoost adaptively chooses η_t as opposed to a fixed η. The theoretical guarantees for AdaBoost are the same, but it performs much better in practice. See Professor Hazan's notes for details.