CS 596: Theoretical Machine Learning

Lecturer: Pranjal Awasthi | Lecture # 8
Scribe: Pengxiang Wu | Oct 13, 2016

In the PAC learning model, our goal is to produce a hypothesis h such that $error(h) \leq \epsilon$, with probability $\geq 1-\delta$. This is a *strong* learning model, since ϵ and δ could be arbitrarily small. Now suppose we are given a set of *weak* algorithms which are good but not perfect, is it possible to combine them to produce a strong and perfect algorithm? This process is called *boosting*, which is the main topic of this lecture.

Intuitively, if the weak algorithms complement each other, we could fuse their respective advantages to achieve a perfect algorithm. More specifically, we will prove the following theorem

Theorem 1 (Boosting)**.** *Given function class H and the target function $h^* : \mathcal{X} \mapsto \{-1,+1\}$, if for some $\gamma > 0$ and any distribution D over $\mathcal{X}$, $\exists h \in H$ such that $err_D(h) \leq \frac{1}{2} - \gamma$, then: (1) for $\forall \epsilon > 0$ and $\forall D$ over $\mathcal{X}$, $\exists g \in MAJ_k(H)$ such that $err_D(g) \leq \epsilon$, where $MAJ_k(H) = \{\mathrm{sgn}(\sum_{i=1}^k \alpha_i h_i(x)); h_i \in H\}$, and $k = O(\frac{1}{\gamma^2}\log(\frac{1}{\epsilon}))$; (2) such g can be found efficiently given access to a weak learning algorithm.*

Here $err_D(g)$ represents true error with respect to the distribution, i.e., $err_D(g) = Pr_{x\sim D}[g(x) \neq h^*(x)]$. The improved function g can be found using AdaBoost, as we will introduce later.

Theorem 1 implies two facts. First, weak learning and PAC learning are equivalent. Of course, a weak learner is not the same thing as a strong learner; by contrast, the error probability of a weak learner is required only to be slightly smaller than half. Here what we mean by "equivalent" is that, a problem can be weak-learned if and only if it can be strong-learned.

Intuitive explanation: Suppose there are three hypotheses $h_1, h_2, h_3 \in H$ that make independent errors probability of p. Then the error rate of their majority vote would be $p^3 + 3p^2(1-p)$. If h_1, h_2, h_3 are produced by a weak learning algorithm (i.e., $p < \frac{1}{2}$), we have $p^3 + 3p^2(1-p) < p$, which indicates an improved performance. One can then bootstrap this process to get down to any arbitrary error rate of ϵ.

Now we give a formal proof of the first part of theorem 1.

Proof. We prove the theorem by setting up and experts problem and using the guarantee of the randomized multiplicative weights (RMW) algorithm. Let $X = \{x_1, x_2, \ldots, x_N\}$ be the set of experts (i.e., examples). On day t, the algorithm picks one expert $x_t \in X$, and the adversary returns a hypothesis $h_t \in H$ accordingly. We define the loss as $1 + h_t(x_t)h^*(x_t)$. Thus when h_t predicts correctly, the loss would be 2, otherwise 0. Let's assume that we are using RMW to pick a distribution over experts at each stage. For a given input probability distribution p_t, the strategy taken by adversary is to find $h_t \in H$ with maximum loss. Now given arbitrary hypothesis h, the expected loss of RMW at time t would be $E[loss(h)] = \sum_{x\in X} p_t(x)(1 + h(x)h^*(x))$. Since, there always exists a weak learner for any distribution p_t, the adversary can always make sure that the expected loss of RMW at each step is at least $1 + 2\gamma$. We also know that on any sequence of length T, RMW guarantees that $E[loss(RMW)] \leq loss\ of\ best\ expert + \sqrt{\dfrac{logN}{T}}$, we have:

$$1 + 2\gamma \leq E\left[loss(RMW)\right] \leq loss\ of\ best\ expert + \gamma \quad [when\ T\ is\ large], \tag{1}$$

where the second inequality comes from the fact that when T is sufficiently large, $\sqrt{\frac{logN}{T}}$ will be smaller than γ (since γ is a small number). Thus it holds that:

$$loss\ of\ best\ expert \geq 1+\gamma,$$

which means:

$$loss\ of\ any\ expert \geq 1+\gamma. \tag{2}$$

Hence we have the following inequality:

$$\begin{aligned} & \frac{1}{T}\sum_{t=1}^{T}(1+h_t(x)h^*(x)) \geq 1+\gamma, \forall x \\ \implies \quad & \frac{1}{T}\sum_{t=1}^{T}h_t(x)h^*(x) \geq \gamma, \forall x \\ \implies \quad & \left[\frac{1}{T}\sum_{t=1}^{T}h_t(x)\right]h^*(x) \geq \gamma, \forall x. \end{aligned} \tag{3}$$

Let $S = \frac{1}{T}\sum_{t=1}^{T}h_t(x)$, and $q_i = \frac{the\ number\ of\ times\ h_i \in H\ appears\ in\ S}{T}$, then it holds that $\frac{1}{T}\sum_{t=1}^{T}h_t(x) = \sum_{i=1}^{|H|}q_ih_i(x)$. Thus $\forall x \in X$, $[\sum_{i=1}^{|H|}q_ih_i(x)]h^*(x) \geq \gamma$. Notice that when $h^*(x) = 1$, $\sum_{i=1}^{|H|}q_ih_i(x) \geq \gamma$; similarly, $\sum_{i=1}^{|H|}q_ih_i(x) \leq -\gamma$ when $h^*(x) = -1$. Next we sample k functions $h_1, h_2, \ldots, h_k$ from distribution $\vec{q} = \{q_1, q_2, \ldots, q_{|H|}\}$. According to Hoeffding's inequality, we have:

$$Pr\left(\left|\frac{1}{k}\sum_{i=1}^{k}h_i(x) - \sum_{i=1}^{|H|}q_ih_i(x)\right| \geq \gamma\right) \leq 2e^{-2k\gamma^2}, \tag{4}$$

where $\sum_{i=1}^{|H|}q_ih_i(x)$ represents the true average of functions. Now fix distribution D over $\mathcal{X}$, and let $g = \text{sgn}(\frac{1}{k}\sum_{i=1}^{k}h_i(x))$, we get:

$$\begin{aligned} err_D(g) &= \sum_x P(x)Pr_g(g\ is\ incorrect\ in\ x\ |x) \\ &= \sum_x P(x)Pr_g\left(\left[\frac{1}{k}\sum_{i=1}^{k}h_i(x)\right]h^*(x) < 0\right) \\ &\leq \sum_x P(x)\cdot 2e^{-2k\gamma^2}, \end{aligned} \tag{5}$$

where $P(x)$ is the probability of picking x. Since we want $err(g) \leq \epsilon$, it follows that $\sum_x P(x)\cdot 2e^{-2k\gamma^2} \leq \epsilon$. Therefore, k should satisfy $k \geq \frac{1}{2\gamma^2}\log\frac{2}{\epsilon}$, i.e., $k = O(\frac{1}{\gamma^2}\log(\frac{1}{\epsilon}))$. $\square$

Next we prove that strong learner g can be found efficiently given an algorithm to find a weak learner for every distribution. Let X be the set of experts, which could be infinite or finite. Now we sample m i.i.d. training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ from D. Let U_m be the uniform distribution over these examples. In the following we will show how to get $g = \text{sgn}(\frac{1}{k}\sum_{i=1}^{k}h_i(x)) \in G = MAJ_k(H)$ such that $\widehat{err(g)} = \frac{1}{m}\sum_{i=1}^{m}I(g(x_i) \neq y_i) \leq \epsilon$.

From VC theorey this would be enough to get small true error as well. Let $d = VCdim(G)$, then we have:

$$err(g) \leq \epsilon + \sqrt{\frac{d \log(\frac{1}{\delta})}{m}}, \; with \; probability \geq 1 - \delta. \tag{6}$$

Therefore, when m goes large, $err(g)$ approaches ϵ. It is worth mentioning that, if $d' = VCdim(H)$, then $VCdim(G) = O(kd' \log(kd'))$.

The way we get g is to run RMW as in the previous case, but only over experts defined by training examples. By the previous analysis, we would be efficiently able to find a g that has low training error. The algorithm that results is known as Adaboost:

Algorithm 1 AdaBoost

Input: m i.i.d. training examples from uniform distribution D
Initialize: $P_1 = (\frac{1}{m}, \ldots, \frac{1}{m})$
for $t = 1, 2, \ldots, T$ **do**
    Get h_t which is a weak learner from P_t
    Let $\epsilon_t = \hat{err}(h_t)$ = error of h_t on the training set
    $P_{t+1}(i) \propto P_t(i) \exp(-\alpha_t)$, if $y_i = h_t(x_i)$
    $P_{t+1}(i) \propto P_t(i) \exp(\alpha_t)$, if $y_i \neq h_t(x_i)$
end for
Output: $g = \mathrm{sgn}(\sum_{t=1}^{T} \alpha_t h_t(x))$, where $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Claim 1. *After T rounds, the training error of g is $\leq e^{-2T\gamma^2}$ in T rounds. After $\frac{2}{\gamma^2} \log(\frac{1}{\epsilon})$ rounds, the training error would be $\leq \epsilon$.*

The above claim states that AdaBoost can find the strong learner g efficiently. However, it is worth mentioning that, if there exists noise in the training data, AdaBoost could be quite bad.

Additional Reading

Chapter 10 from the book by Shai Shalev-Shwartz and Shai Ben-David. See online copy: `http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf`